Summer 1978

An Investigation into a Least Squares Method for Image Registration

Ernest William Cordon
University of Central Florida, bcordan@sprynet.com

AN INVESTIGATION INTO A LEAST SQUARES METHOD FOR IMAGE REGISTRATION

BY

ERNEST WILLIAM CORDAN, JR.
B.S.E.E., MEMPHIS STATE UNIVERSITY, 1974

THESIS

Submitted in partial fulfillment of the requirements
for the degree of Master of Science: Engineering
in the Graduate Studies Program of
the College of Engineering
of Florida Technological University at Orlando, Florida

Summer Quarter
1978

AN INVESTIGATION INTO A LEAST SQUARES METHOD FOR IMAGE REGISTRATION

BY

ERNEST WILLIAM CORDAN, JR.

ABSTRACT

One of the problems associated with the automatic image processing of satellite photographs such as weather maps is the need for image registration; that is, the fitting of a map that has some translational and rotational bias to a known data base. This paper investigates a least square method of image registration using an image that has been converted into a boundary map with a pixel representation of 1 for land, -1 for water and zero for cloud pixels. A sampled correlation array is constructed by shifting the weather map to locations on a given grid, centered around a sampled correlation peak, and performing an accumulation of the pixel-by-pixel comparisons between the weather map and its data base over the whole map or a smaller search window. A least square approximation of the translational and rotational bias is performed using the data from this sampled correlation array, fitted against the shape of an elliptical cone.

ACKNOWLEDGMENTS

The author is indebted to Dr. Benjamin W. Patz for his invaluable counsel throughout this project and would also like to express his gratitude to the management of the Hybrid Computational Sciences Department of Martin Marietta Aerospace, Orlando Division, for the use of their computational equipment during this project.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

I. INTRODUCTION

One of the problems associated with the automatic image processing of incoming satellite maps such as weather map photographs, military reconnaissance photographs, or earth resource maps is the need for image registration; that is, the fitting of a map that has some translational and rotational bias to a known data base. Therefore it is necessary to perform a pixel-by-pixel comparison of the photograph with the stored data base and using this comparison data it should be possible to perform a spacial registration of the photograph removing any translational and rotational bias associated with the image. This spacial registration must be performed despite the introduction of noise or distortions on the image. Geometric distortions and magnification differences may be adjusted by performing adjustments to the data base knowing the position of the satellite.

This paper considers the problem of registering the translational and rotational differences of an image using the weather map photograph as the subject. These shifts should be registered despite the noise introduced by clouds in the picture. The image is assumed to have been converted into a boundary map with a pixel representation

of 1 for land, -1 for water and zero for cloud pixels. The computer data base, however, contains only elements of 1 and -1 to represent its land and sea area respectively.

II. BACKGROUND

Misregistration occurs when image sensings are separated in time and space such that spacial alignment of the sensor is impractical or impossible. The process of aligning the sensed image with a reference image by translation and rotation is called registration [1]. Translation is that distance along two orthogonal axes which a point on the reference image must undergo to be aligned with the corresponding point on the sensed image. Rotation is that augular distance which a point in the reference image must undergo to have the same augular orientation as its corresponding point in the sensed image.

One of the earliest methods of image registration is that of cross-correlation. The continuous correlation function of x and y can be expressed in two dimensions as

$$\gamma(k,l) = \int\int_{-\infty}^{\infty} x(r,s)y(r-k,s-l)\,dr\,ds \quad . \qquad (1)$$

Since correlation in the time or space domain is equivalent to multiplication in the frequency domain (the transform of x(r,s) by the complex conjugate of the transform

of y(r,s)), Fourier transforms can be used to determine an image correlation. The use of a fast Fourier transform (FFT) algorithm leads to a high speed correlation method [2] which has been used in image registration [3,4].

Barnea and Silverman [5] developed a set of sequential similarity detection algorithms (SSDA) that determine the translational misregistration by as much as two orders of magnitude faster than FFT correlation. These algorithms search for the correlation peak and reject correlation samples that do not meet a certain threshold. SSDA's have been combined with an iterative least squares estimation to perform a translational and rotational registration [6].

Translation and rotation can be registered by high speed cross-correlation if it is performed twice, once in cartesian coordinates and again in polar coordinates, since a rotational shift is equivalent to a translational shift in polar coordinates [7]. This paper proposes to perform a sampled correlation [8], followed by a least squares approximation of the translational and rotational misregistration.

III. APPROACH

The weather map (this paper uses a weather picture as the misregistered image) is stored in the computer's memory as a three level map containing two component levels (land and sea) with values of 1 and -1 and a disturbance level (clouds) with a value of zero. The image is assumed to be free of geometric distortions and magnification error with respect to a reference image stored in a data base. The coordinate systems of both the weather map and data base map have their origins located at the center of the map. The coordinates of the weather map (u',v') are related to the data base coordinates (u,v) by the linear transformation,

$$\begin{aligned} u' &= u\cos\theta - v\sin\theta + \alpha \\ v' &= u\sin\theta + v\cos\theta + \beta \end{aligned} \quad , \tag{2}$$

where α and β is the translational misregistration along u and v respectively and θ is the angular rotational misregistration of the weather map image.

A sampled two-dimensional correlation is performed between the weather map image and the data base. This is done by shifting the weather map along the u- and v-axes

in equal-distant increments and performing an accumulation of the pixel-by-pixel comparisons over the whole picture or search window at each shift point. This comparison is defined as

$$R(u,v) = \sum_{j=-J}^{J} \sum_{k=-K}^{K} M_w(j-u,k-v)\, M_d(j,k) \qquad (3)$$

where j and k are pixel indices in a 2J by 2K search window area; M_w and M_d are the weather map and data base map respectively. Whenever like pixels are compared, 1 is added to the accumulation and whenever unlike pixels are compared, -1 is added to the accumulation. If a cloud pixel is detected, the accumulation is left unchanged, since clouds represent unusable portions of the weather map for purposes of registration and are therefore masked out of the summation.

The sampled correlation is then searched to locate the peak sample. This peak sample is separated from the continuous correlation peak by less than half the sample grid in u and less than half the sample grid in v.

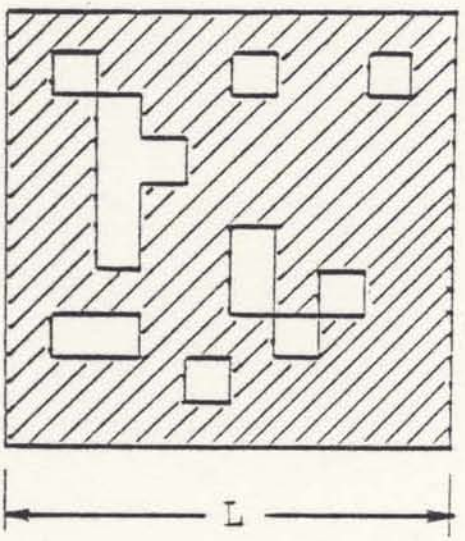
A square sampled correlation array is obtained from a set of samples around and including the peak sample. The peak sample serves as the center element of the array. In this paper, a 5 x 5 correlation array was constructed

by taking the peak sample and all samples within ±2 samples of the peak in both the u- and v-direction. The samples were spaced 20 pixels apart and had coordinates of $(u_{pk} \pm 20n, v_{pk} \pm 20m)$, where n and m are integers from 0 to 2 and (u_{pk}, v_{pk}) represents the coordinates of the peak sample.

Once the sampled correlation array has been established, the sample elements of the array and their corresponding coordinates are entered into a least squares approximation to estimate the translational and rotational bias on the weather map. This translational and rotational result will be with respect to the location of the peak sample. Any translational bias on the picture will be no more than half the sample grid in v since the array is centered about the sampled correlation peak.

The distance between sample points, when constructing the correlation array, is determined by the smallest detail on the map which is to be included in the registration process [9,10]. However, if very small details such as a salt-and-pepper pattern, were to be included in the registration process, then the pattern could be interpreted as noise.

Figure 1 shows the correlation in the u-direction (at v equal zero) of a square island containing many small



Island Map

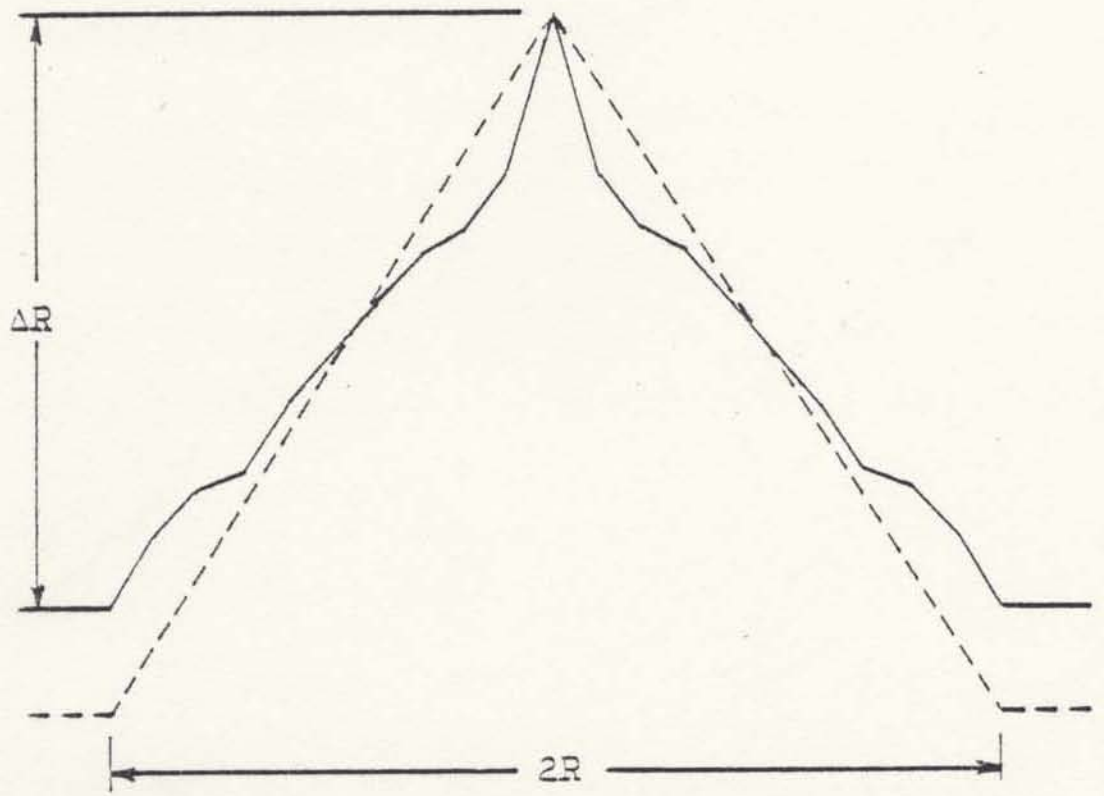


Figure 1. Correlation of a square island containing lakes (solid line) versus that of the same island without lakes (dashed line)

lakes. The correlation of this pattern (solid line) appears to be noisy when compared to that of the same island containing no lakes (dashed line). The slope of the correlation in figure 1 is distorted by the lakes when they match against other lakes adding to the correlation (increasing the slope) and when they mismatch against land regions subtracting from the correlation (decreasing the slope).

The change between minimum and maximum correlation, ΔR, is smaller than that of the island containing no lakes, because when the island is shifted over the larger water regions during the correlation process, these small lakes match, adding to the correlation value. If the detailed pattern were severe enough, it would be possible for the correlation peak to be masked out altogether. It may be necessary to reclassify this detail such that the reclassified regions become "large" compared to the sampling distance over the correlation function [11].

IV. IMAGE CORRELATION

To generate the data necessary for a least square image registration, the received map must be correlated against a stored data base map. A correlation is achieved using the process described by equation (3) in which pixels of the weather map are compared with pixels of the stored data base map and the function, R(u,v), is increased by 1 for comparisons of like pixels, -1 for comparisons of unlike pixels and zero for comparisons of cloud pixels. Figure 2 illustrates how matched, mismatched and cloud regions provide the components for R(u,v) when a weather map is compared to the data base. The data base is assumed to be large enough so that the weather map, when shifted and rotated, would never exceed the size of the data base.

Correlation of a Simple Map

To illustrate the shape of R(u,v) as it is effected by a translational or rotational bias on the weather map, a simple map of 1500 x 1500 pixels is considered containing one square island with dimensions of 240 x 240 pixels. The correlation is performed by shifting the map over the data base in the u- and v-directions as described

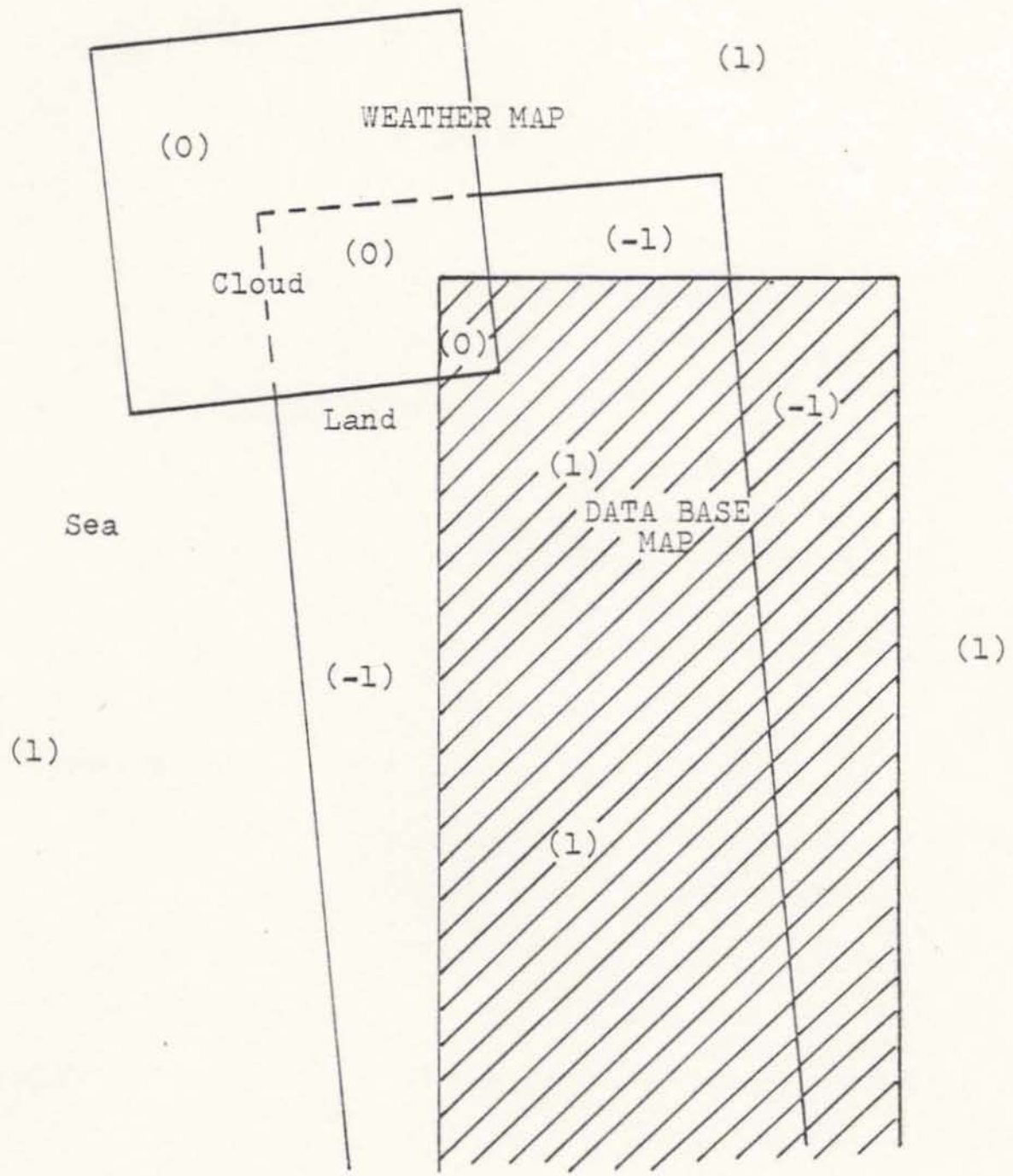


Figure 2. Components of R(u,v) after comparison with data base

by equation (3). The function, R(u,v), rises from a minimum when the entire weather map island is over the data base's water region. In this position, -1 is added to R(u,v), for each pixel of the weather map's or data base's island that compares to the data base's or weather map's water region and 1 is added to R(u,v) for each of the remaining pixels in the weather map's water region which compare with the data base's water region. This minimum has a value of

$$[(240)(240)](-1) \quad \text{weather map island}$$
$$+[(240)(240)](-1) \quad \text{data base island}$$
$$+[1500^2 - 2(240^2)](1) \quad \text{remaining weather map pixels}$$

or 2,019,600. A maximum correlation is achieved when a maximum amount of weather map land region covers the data base island region. If the weather map had no translational or rotational bias, then both islands would exactly overlap and R(0,0) would be equal to [1500 x 1500](1) or 2,250,000.

Figure 3 illustrates the continuous correlation function, R(u,v), for a weather map containing a square island with no translational or rotational bias. Another view of this correlation is shown with the contour-map in figure 4. The surface of this correlation, along the u- and v-axes, falls from the peak with slopes of -960 as

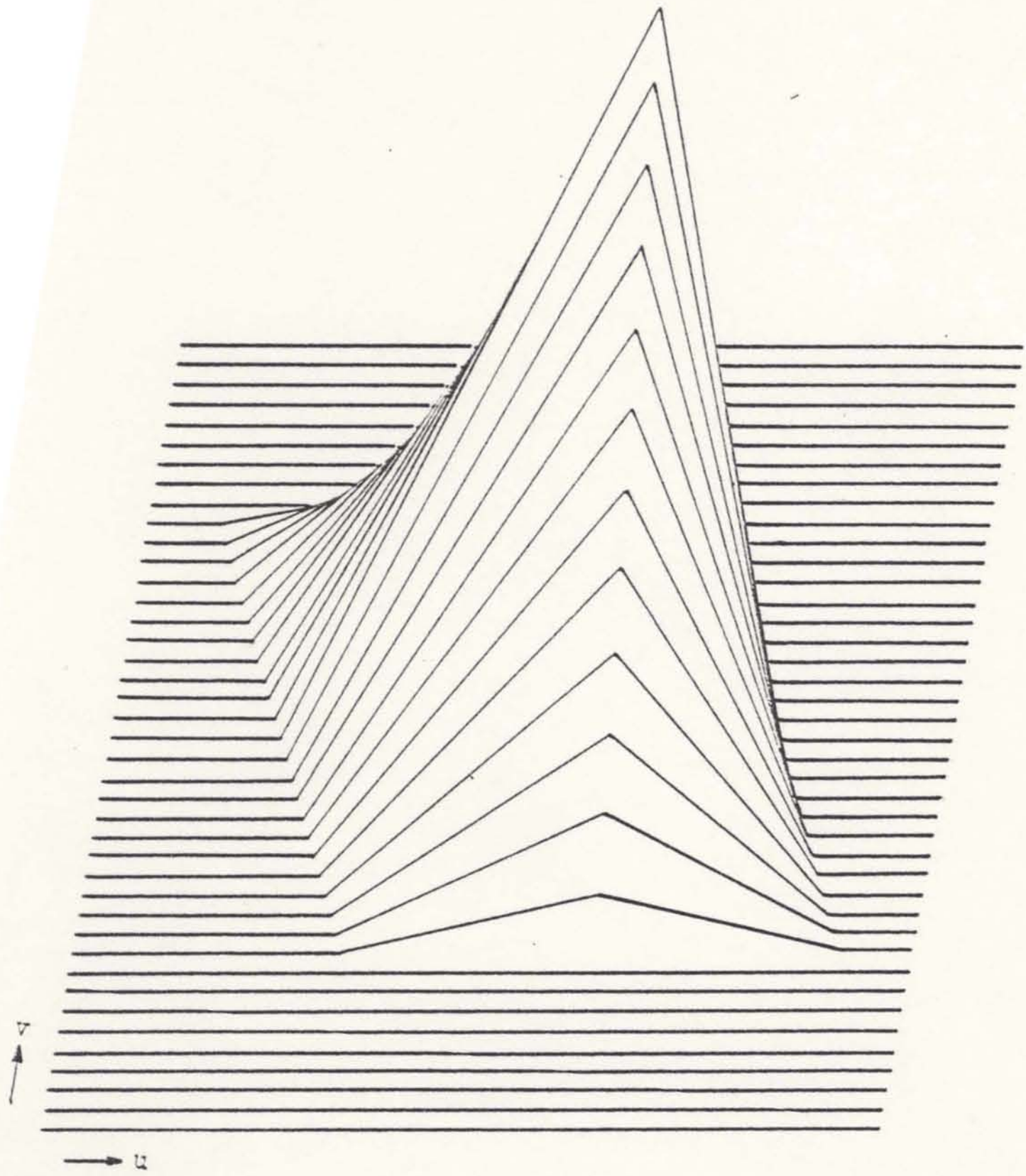


Figure 3. Correlation of a square island, $R(u,v)$

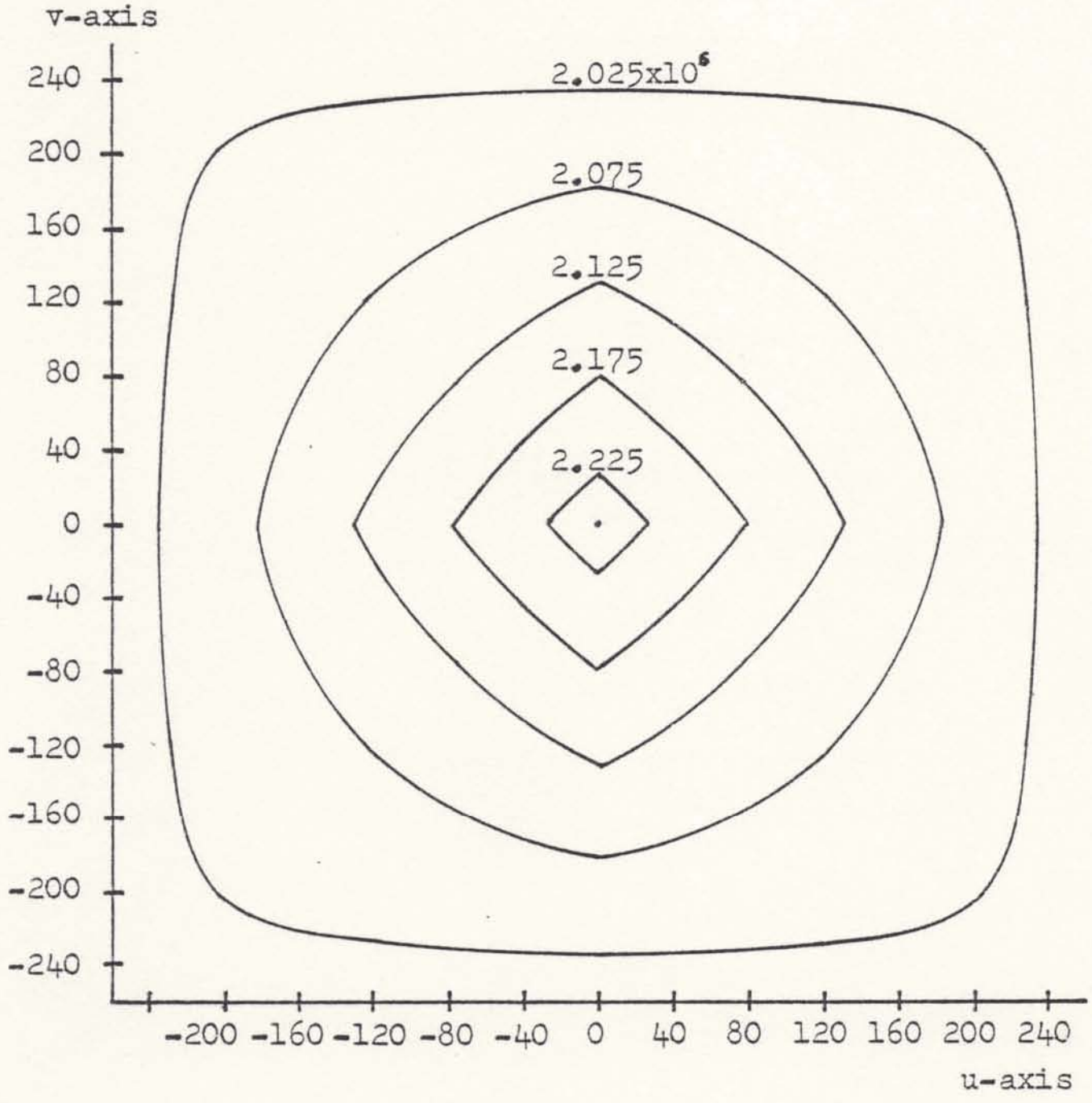


Figure 4. R(u,v) for a square island with no rotation

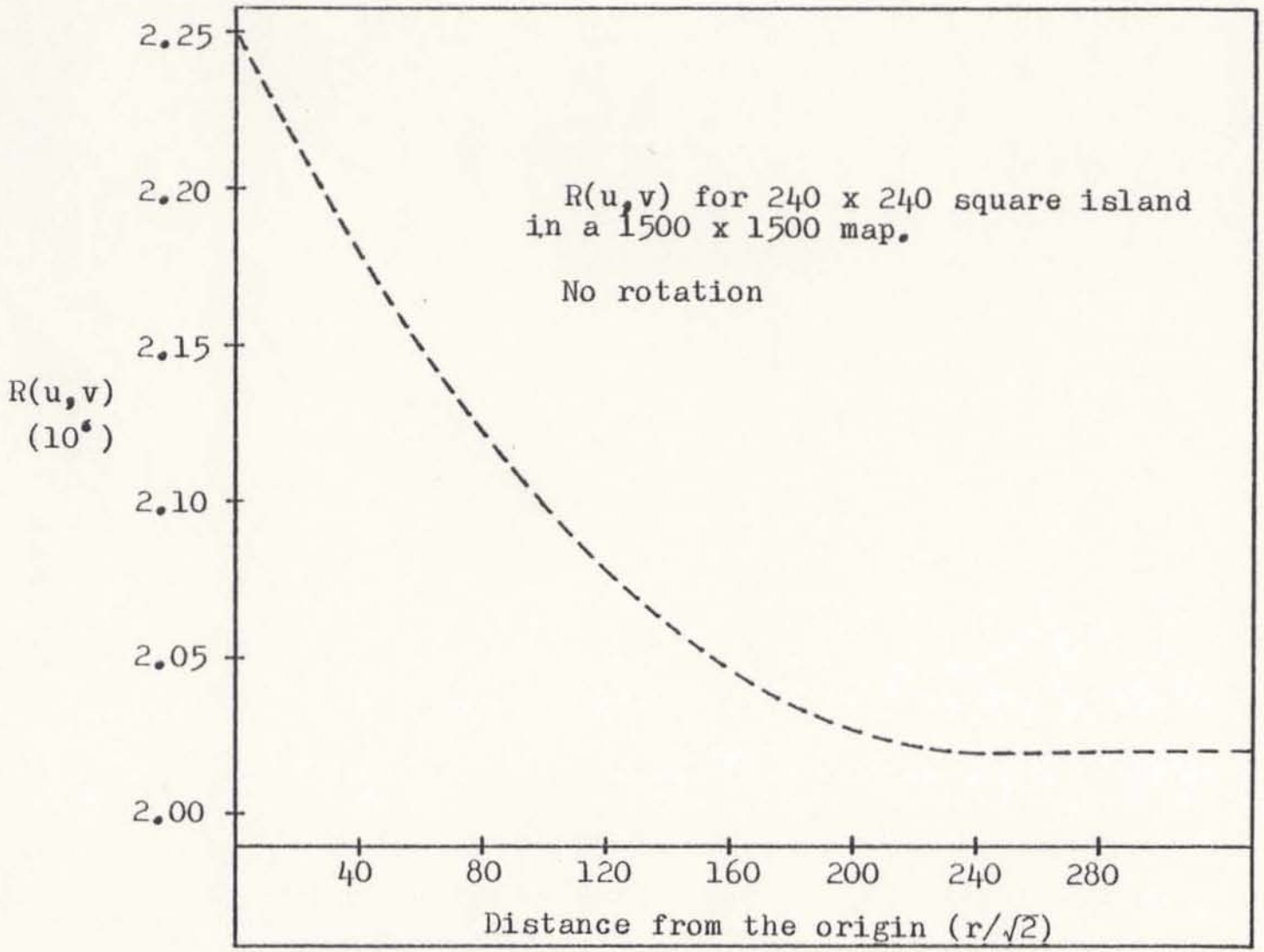


Figure 5. Cross section of R(u,v) for a square island at the line |u|=|v|

the map is shifted along these axes. These surfaces meet to form a curved edge at an angle of 45° from either the u- or v-axis as shown in the correlation cross section illustrated in figure 5.

To illustrate the effect of rotation on the map correlation, the simple map containing a 240 x 240 square island is correlated with a rotational bias on the weather map of 45°. The shape of this correlation is shown with a contour-map in figure 6. In this case of 45° rotation, the continuous correlation rises from an octagonal shaped base and changes quickly into a circular shape. Figure 7, which shows a diagonal cross section at 45° from either the u- or v-axis, indicates that the peak of this correlation is blunt when compared to that of the correlation of a map with no rotational bias (figure 5). This lower peak correlation value is due to the fact that the map cannot exactly overlap the data base when it is rotated and the corners of both weather map and data base map islands will cover water regions causing mismatching which decreases the correlation value.

A translational bias on the weather map would not affect the shape of the correlation in any way except to shift the peak's location with respect to the center of the weather map. If, for example, the weather map has a

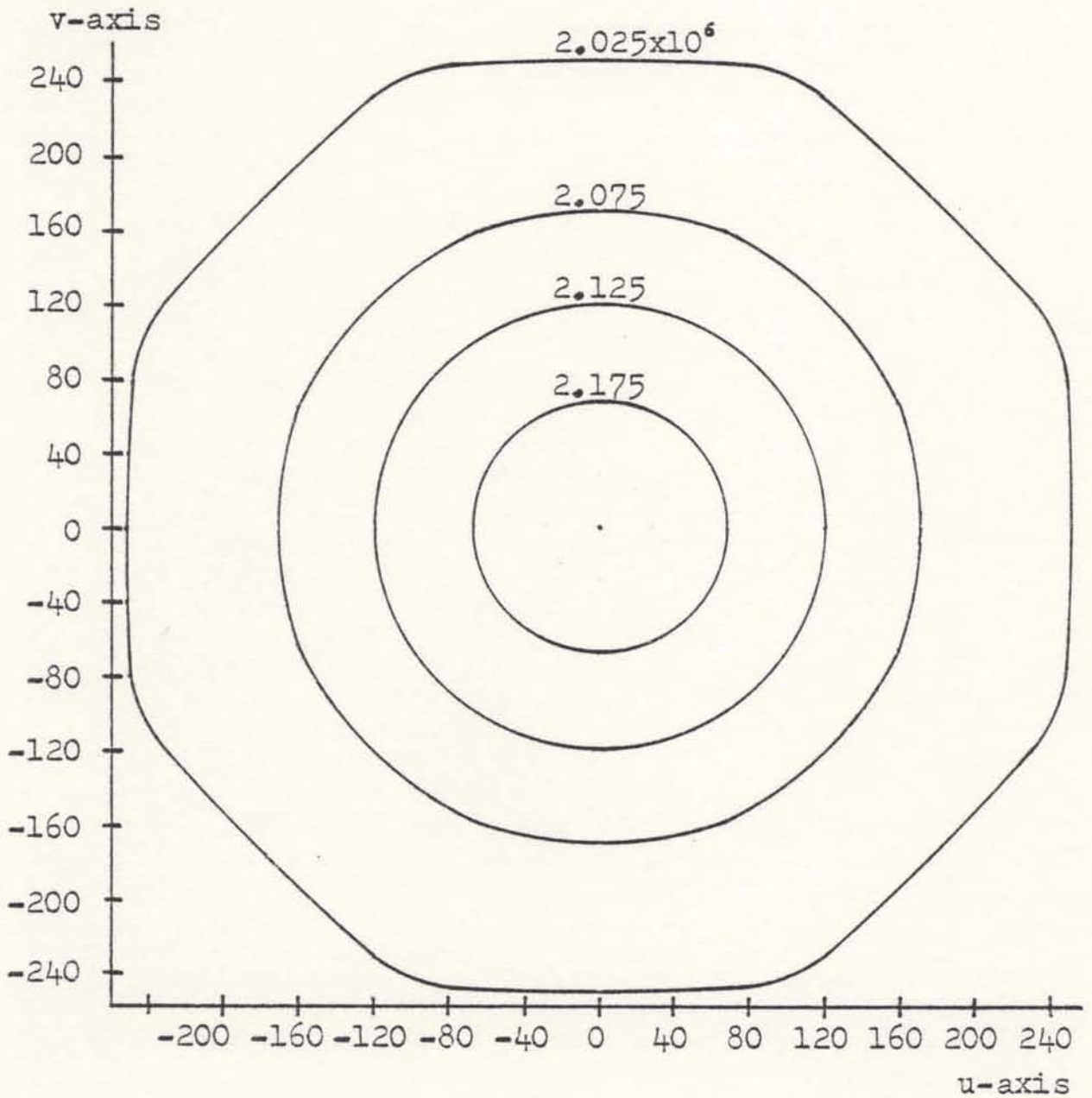


Figure 6. R(u,v) for a square island with 45° rotation

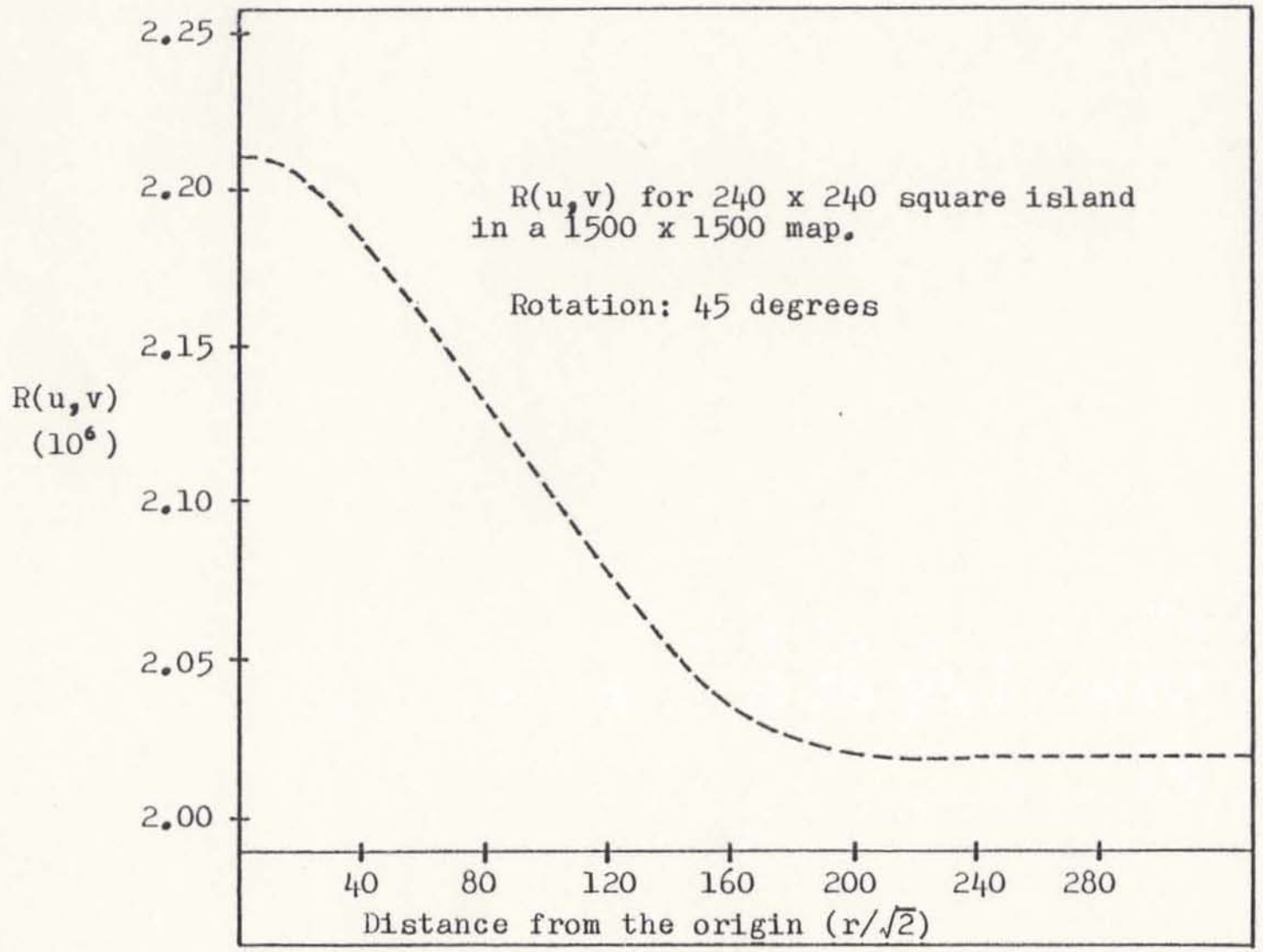


Figure 7. Cross section of R(u,v) for a square island at the line |u|=|v|

tranlational bias of α and β with respect to the data base, then the continuous correlation peak would be shifted $-\alpha$ and $-\beta$ with respect to weather map center.

The next step towards a more general weather map model would be the consideration of a rectangular island. Throughout this paper, a hypothetical map model will be used containing an island of 900 x 210 pixels within a 1500 x 1500 pixel weather map. Also a square cloud of 100 x 100 pixels may be placed anywhere on the map. Figure 8 illustrates this hypothetical map.

The shape of a rectangular island's correlation without clouds follows the same pattern as that of the square island except that the rectangular island's correlation will be elongated toward the direction of the island's longer side, which for the hypothetical map model would be in the vertical direction or the v-axis. As the weather map model is rotated, then the continuous correlation contour-map would tend towards concentric ellipses. In other words, the correlation would vary more slowly in the v-direction or

$$\left|\frac{\partial R}{\partial v}\right| < \left|\frac{\partial R}{\partial u}\right| \qquad (4)$$

due to the longer proportions along the v-axis. As the

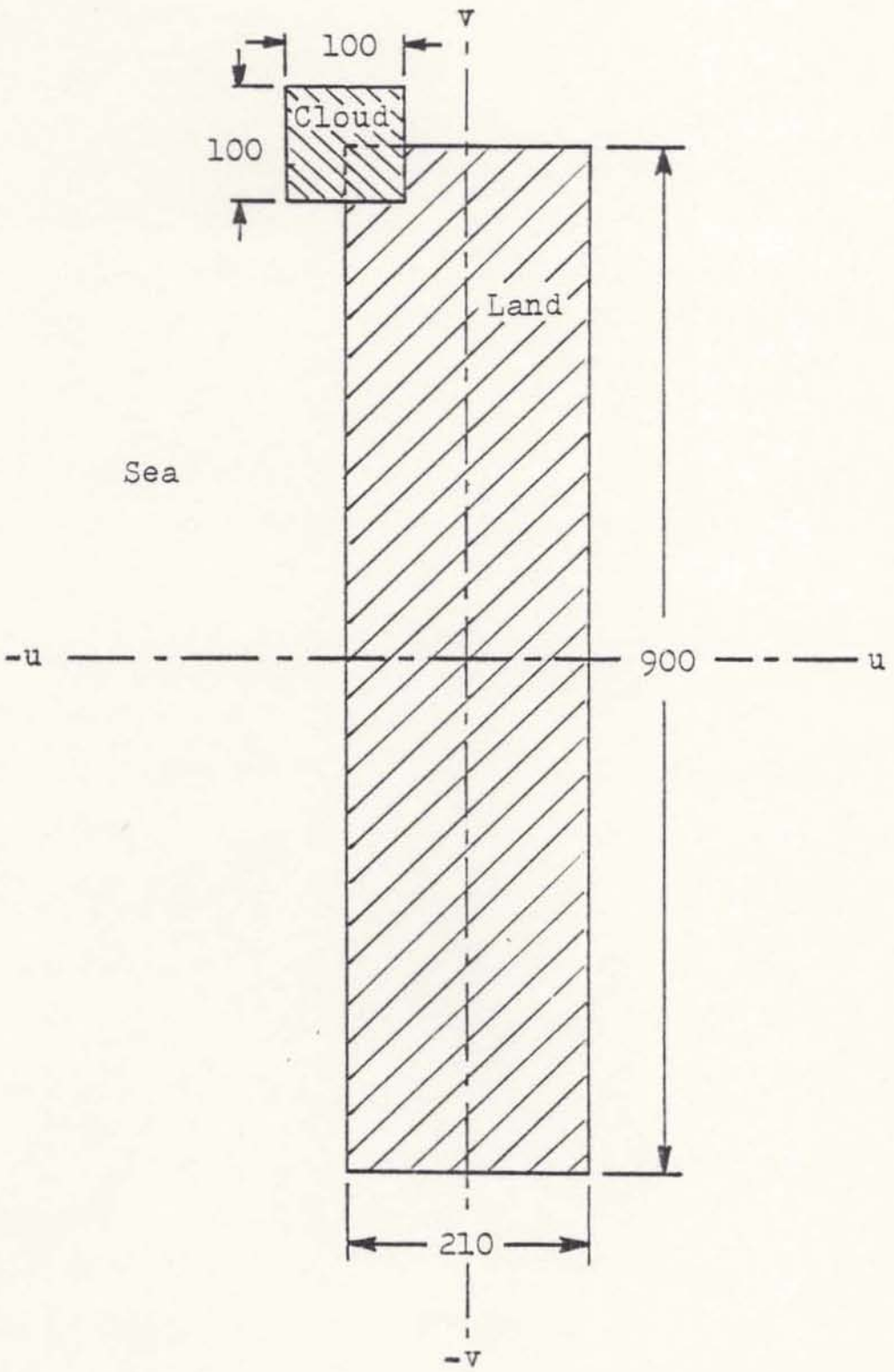


Figure 8. Hypothetical map model

weather map image approaches a more realistic shape, the correlation surface would no longer have sharply defined edges nor would the surface slopes be constant.

The Sampled Correlation Array

After the sampled correlation has been performed and the peak sample has been located, a square correlation array is obtained from a set of samples around and including the peak sample. For the purpose of simulating the least squares registration method, it was assumed the sampled correlation peak was found to be at the origin bringing the continuous correlation peak within ±10 pixels along u and ±10 pixels along v. A 5 x 5 sampled correlation array had to be generated around the sampled peak (origin) with a sample grid spacing of 20 pixels. The center element of this array would be the sample peak value and the other elements (sample values) would be located at coordinates of (±20n, ±20m) where n and m are integers with values of 0 and 2. It should be noted that, ordinarily these elements are available from the sampled correlation from which the peak sample was located; however, in this simulation, only 25 elements need to be determined.

Each element of the simulation's correlation array is generated by shifting the map model to each of the 25 grid

locations and performing a pixel-by-pixel comparison with the data base and accumulation using equation (3), where like pixel comparisons add 1, unlike pixel comparisons add -1 and cloud pixels add zero to the accumulation. After all of the weather map model's pixels have been compared with the data base, the resulting accumulation is entered as a sample on the correlation array's grid. The weather map model is then shifted to a new location on the grid and the entire pixel-by-pixel comparison process is repeated until all 25 grid samples have been calculated.

The FORTRAN program used for the generation of the sampled correlation array (listed in Appendix 2) follows the procedure outlined in the previous paragraph. If the clouds in the weather map model are placed near the island, then a large area of water that surrounds the land and cloud region will compare with the water surrounding the data base island. This water-to-water comparison will add a value of 1 for each weather map pixel of this outside region; therefore a search window is established that excludes the outside water region, and the pixel comparison process is applied within the search window only. Once all the pixels within the search window have been processed, the accumulation

is then added to the area of the outside region to yield the total sample value of the particular grid location being processed. Whenever the weather map model is shifted to a new grid location, a new search window is established to include all land and cloud regions of the weather map and data base map.

This search window is established by examining the coordinates of all the corners of the islands (weather map and data base map) and clouds and finding the maximum absolute value of the corners' u-coordinates and the maximum absolute value of the corners' v-coordinates. These two maximums make up the coordinates of the search window's corner in the first quadrant and their reflections make up the corners of the search window in the other three quadrants. In other words, if (u_i, v_i) are the coordinates of the corners of the weather map island, data base island and cloud, then the search window is established between

$$u_w = \pm(\max|u_i|) \text{ and } v_w = \pm(\max|v_i|) \quad , \tag{5}$$

for all i, and the window's area would be

$$\text{Area} = 4|u_w v_w| \quad . \tag{6}$$

The window's area would be the only area in which it is

necessary to perform a pixel scan and comparison, thereby reducing the time needed to determine the value of each sample. Figure 9 further illustrates how the search window is established based on a hypothetical weather map model and its data base.

In obtaining data for this paper, the search window's area ranged from 189,000 pixels to 737,000 pixels which when compared to a total weather map area of 1500 x 1500 or 2,250,000 pixels represented a scan area reduction of 92% to 67%. Nevertheless, the Interdata 8/32 system used in this project required anywhere from 20 minutes to one hour for the generation of a 5 x 5 sampled correlation array.

The generation of the correlation array may be done more efficiently if the computer used has an architecture tailored to this particular image processing task. The throughput of a specialized image processing computer could be orders of magnitude faster than that of a general purpose computer such as the one used to provide data for this paper [12]. A cellular-logic-array processor like the Staran parallel processor, which is an array of simple processors each of which is assigned a particular data stream, has an architecture that is well suited for many image processing applications [13,14]. Since each pixel

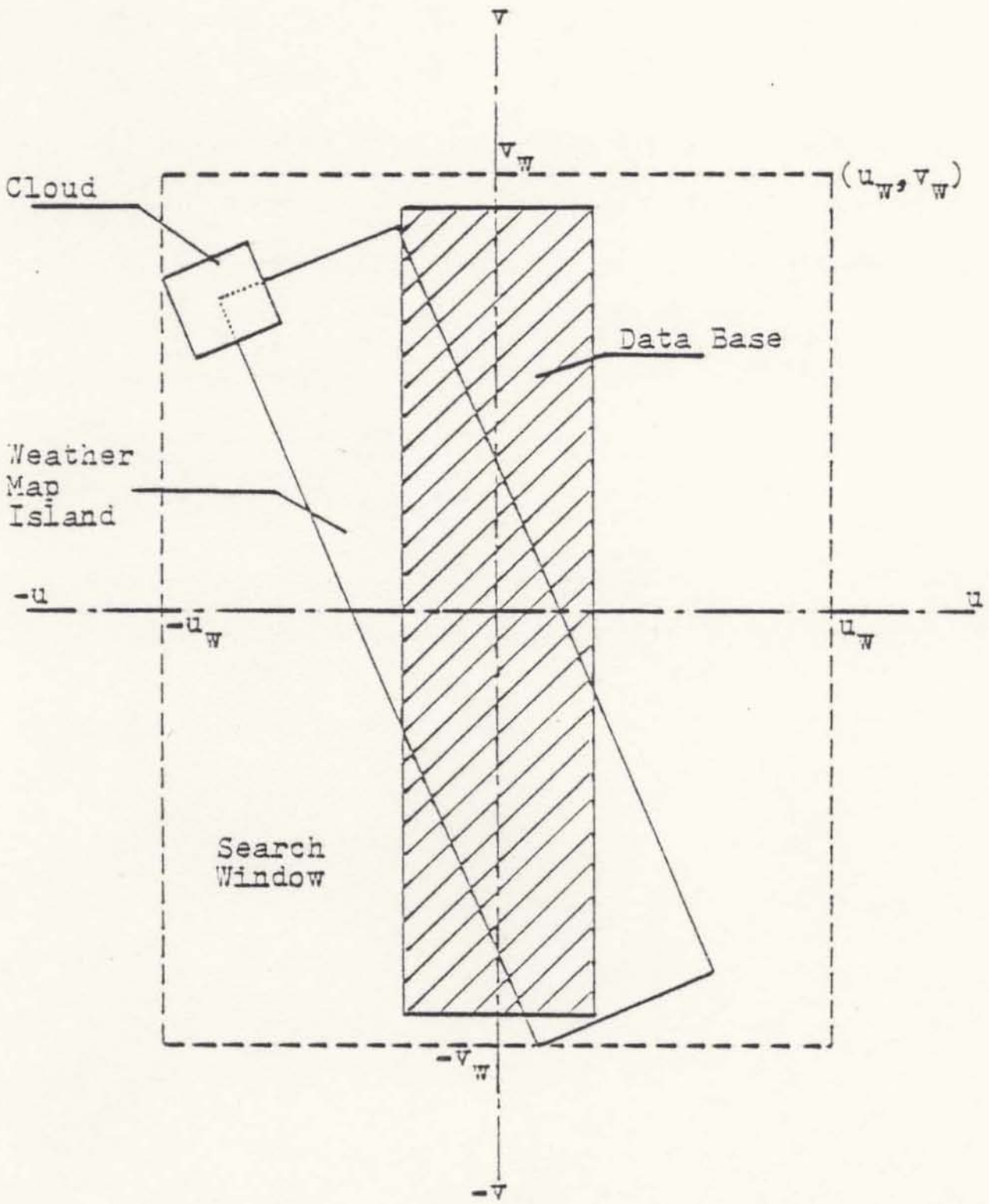


Figure 9. Search window construction

is computed by the same algorithm, they can be processed in parallel by this type of single-instruction-stream multiple-data-stream architecture. Due to the technological advances that have been made with LSI processing units such as microprocessors, it is becoming more economical to construct large, cost-effective parallel image processors with architectures similar to that of the Staran computer.

V. LEAST SQUARE REGISTRATION METHOD

The least square approximation method is used to fit physical data to a chosen function. To chose a function for this application, it is necessary to examine the shape of the continuous correlation of a weather map against its data base. The square island correlation shown in figure 3 and its contour map shown in figure 4 suggest that a pyramid might be used to approximate the correlation for least squares fitting. The pyramid, however, has two disadvantages; one, it has edges or discontinuities which would increase the complexity of the equations used in a least squares fit and, two, it has flat surfaces that do not approximate the correlation as the weather map is rotated (also, the edges of the square island correlation tend to dissolve during rotation). The edges and flat surfaces of the pyramid would be less appropriate for more realistic land formations which generally do not have straight line borders or sharp corners.

Figure 6, concerning the correlation of a square that has a rotational bias, suggests a conical shape. The linearity along the u- and v-axis of the nonrotated

square's correlation and the linearity shown in a large part of the rotated cross section in figure 7 tends to reject such quadratic shapes as the paraboloid, spheroid and hyperboloid in favor of the cone.

A cone with an elliptical base would favor the rectangular island correlations with different proportions in the u-v plane of the correlation. Therefore, the elliptical cone was used for this application of the least squares approximation method. Since the map is to be registered in translation as well as rotation, the cone equation,

$$[h(x,y)]^2 = [x-\alpha]^2/a^2+[y-\beta]^2/b^2 \quad , \tag{7}$$

where α and β are the u- and v-directional translations respectively, is rotated in θ yielding

$$\begin{aligned}[h(x,y)]^2 = {} & [x\cos\theta - y\sin\theta - \alpha]^2/a^2 \\ & +[x\sin\theta + y\cos\theta - \beta]^2/b^2 \quad .\end{aligned} \tag{8}$$

After some algebraic manipulation the cone equation becomes

$$\begin{aligned}[h(x,y)]^2 = {} & [\cos^2\theta/a^2+\sin^2\theta/b^2]x^2 \\ & +[\sin^2\theta/a^2+\cos^2\theta/b^2]y^2 \\ & +2\sin\theta\cos\theta[1/b^2-1/a^2]xy \\ & -2[\alpha\cos\theta/a^2+\beta\sin\theta/b^2]x \\ & +2[\alpha\sin\theta/a^2-\beta\cos\theta/b^2]y \\ & -[\alpha^2/a^2+\beta^2/b^2]\end{aligned} \tag{9}$$

or

$$[h(x,y)]^2 = (t_1)x^2+(t_2)y^2+(t_3)xy+(t_4)x+(t_5)y+(t_6) \quad . \tag{10}$$

The coefficients, t_n, which are obtained by means of least squares are then used to determine α and β , the translational shift, and θ, the rotational component [see appendix 1].

To simulate the least square method of picture registration, the hypothetical map model shown in figure 8 was used. The sampled correlation array was constructed using a 5 x 5 grid centered about the origin of the data base map with a grid spacing of 20 pixels. A translational and/or rotational bias was introduced to the weather map model with the translation being within ±10 pixels in the u-direction and within ±10 pixels in the v-direction. The correlation array has been centered at the sampled correlation peak causing any translational bias to be less than half the sampled grid in u and v.

The normalized elements of the 5 x 5 sampled correlation array, the (u,v) coordinates of the sampled elements and the coefficients of equation (10) yield a set of 25 equations in 6 unknowns, of the form

$$[h^2] = [u^2, v^2, uv, u, v, 1] \begin{bmatrix} t_1 \\ t_2 \\ t_3 \\ t_4 \\ t_5 \\ t_6 \end{bmatrix} \tag{11}$$

or

$$[h^2] = \begin{bmatrix} 1600 & 1600 & 1600 & -40 & -40 & 1 \\ 400 & 1600 & 800 & -20 & -40 & 1 \\ & & \vdots & & & \\ 400 & 1600 & 800 & 20 & 40 & 1 \\ 1600 & 1600 & 1600 & 40 & 40 & 1 \end{bmatrix} \begin{bmatrix} t_1 \\ t_2 \\ t_3 \\ t_4 \\ t_5 \\ t_6 \end{bmatrix} \tag{12}$$

where h^2 is defined as the normalized correlation element,

$$h^2 = [H - R(u,v)]^2 / H^2 \tag{13}$$

normalized by the constant,

$$\begin{aligned} H &= (P_l + P_w) - P_c \\ &= (1500^2) - P_c \quad , \end{aligned} \tag{14}$$

where P_l and P_w are the total number of possible land and water pixels respectively and P_c is the number of cloud

pixels present on the map.

Equation (11) and (12) form the matrix equation

$$[h^2] = w = At \tag{15}$$

which can be solved by

$$A^T w = A^T A t \tag{16}$$

$$[A^T A]^{-1} A^T w = [A^T A]^{-1} [A^T A] t \tag{17}$$

$$[A^T A]^{-1} A^T w = t \quad , \tag{18}$$

the least square solution of t [15]. The matrix $[A^T A]^{-1} A^T$ can be stored as a constant matrix if the sample grid remains unchanged from picture to picture. However, if the grid spacing should vary, then a matrix inversion must be performed each time. This matrix inversion is simplified by the fact that $[A^T A]$ is a symmetric matrix [16]. Appendix 3 contains the FORTRAN least squares program used for this paper.

VI. RESULTS

Using the hypothetical weather map model shown in figure 8, sampled correlation arrays were generated with weather map translational biases of $0, \pm 2.6, \pm 5.2$ pixels for α and β and rotational biases between 0 and 22.5 degrees. These values were combined to provide cases for examination with or without cloud noise. The elements of these correlation arrays were entered into the least square method and solutions were obtained for the translational and rotational biases originally applied to the weather map model (Appendix 4 lists the arrays and their least square results). Table 1 lists the originally applied biases versus their least square solutions for translated and rotated maps containing no clouds. A relative error (in percent) is also given of the least square solutions with respect to the applied biases.

When one examines case 1 of table 1, the case of zero shift and rotation, it can be seen that no zero bias error exists in this procedure. In other words, with zero shift and rotation, the least square method yielded an exact solution offering no bias of its own. This result, however, is not totally unexpected since,

CASE #	ORIGINAL BIAS			LEAST SQUARE RESULTS					
	α	β	θ	α	|α| (%) ERROR	β	|β| (%) ERROR	θ	|θ| (%) ERROR
1	0	0	0	0	0	0	0	0	0
2	0	0	2.5	0	0	0	0	1.330	-46.8
3	0	0	5.0	0	0	0	0	3.174	-36.5
4	0	0	7.5	0	0	0	0	4.876	-35.0
5	0	0	10.0	0	0	0	0	6.356	-36.4
6	0	0	22.5	0	0	0	0	11.874	-47.2
7	5.2	2.6	0	4.780	-8.1	2.300	-11.5	0.122	-
8	2.6	5.2	0	2.848	9.5	4.005	-23.0	0.119	-
9	2.6	-5.2	0	2.848	9.5	-4.005	-23.0	0.119	-
10	-5.2	2.6	0	-4.780	-8.1	2.300	-11.5	0.122	-
11	5.2	2.6	2.5	5.207	0.1	1.811	-30.4	1.420	-43.2
12	5.2	2.6	7.5	5.858	12.7	1.611	-38.0	4.978	-33.6
13	2.6	5.2	7.5	3.302	27.0	3.977	-23.5	4.975	-33.7
14	5.2	2.6	22.5	5.522	6.2	0.744	-71.4	11.879	-47.2

Table 1. Least square results for noncloud maps

in this case, the convolution array has symmetry about the u- and v-axes like an elliptical cone.

Cases 2 through 6 show the results of placing only rotation on the maps. Table 1 illustrates that, in these cases, when no translation has been introduced, the least square method gave an exact result for the translational biases. However, an average error in θ of approximately 40% was introduced to the rotational result.

Cases 7 through 10 show the effect of a pure translational bias in different quadrants. The solution of pure translation showed average errors of ±8.8% in the u-direction and approximately 17% in the v-direction. The increased error in the v-direction, which is predominant throughout this investigation, relates to the shape of the land which was longer in the v-direction causing a lower correlation slope along the v-direction or

$$\left|\frac{\partial R}{\partial u}\right| > \left|\frac{\partial R}{\partial v}\right| \quad . \tag{19}$$

When translations were applied to reflected points in other quadrants there was no change in the least square results. This is seen by inspection of case 7 vs. case 10 and case 8 vs. case 9. A small rotation error was

introduced by the least square approximation of these nonrotational cases. These values were small because the translation bias is uniform throughout the map.

All the results of table 1 were based upon a normalization factor on the coefficients of equation (10) of 2,250,000 which is the maximum possible correlation value of a cloudless map of 1500 x 1500 pixels as calculated with equation (14). A study of the effect of normalization was made on cloudless maps with rotations of 22.5 degrees or less. In figures 10,11 and 12 a plot of the least square solutions versus normalizations from 100,000 to 7,500,000 show that a pole exists at some point within this interval and from there the solution slopes asymptotically toward a solution that is within ±7% of half the desired value (Appendix 5 tabulates calculated rotation vs. the normalization factor). One convenient normalization point is that which is equal to the maximum value of the correlation array (designated by point "a" in the figures). Its solution is closer to the pole and apparently closer to the desired solution, therefore, it appears to be a better approximation.

Table 2 shows the results for the same shift and

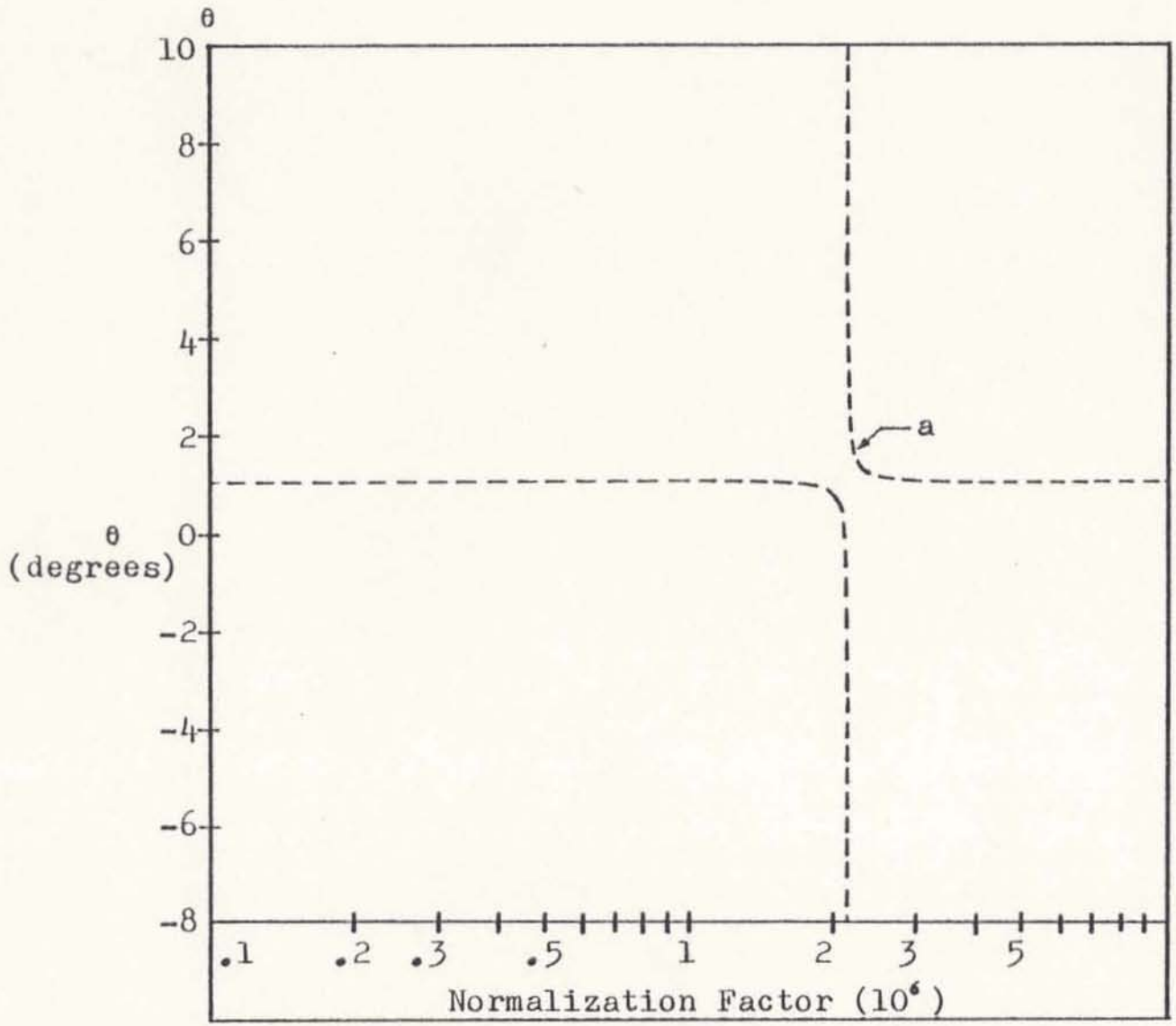


Figure 10. Normalization of $\theta = 2.5°$

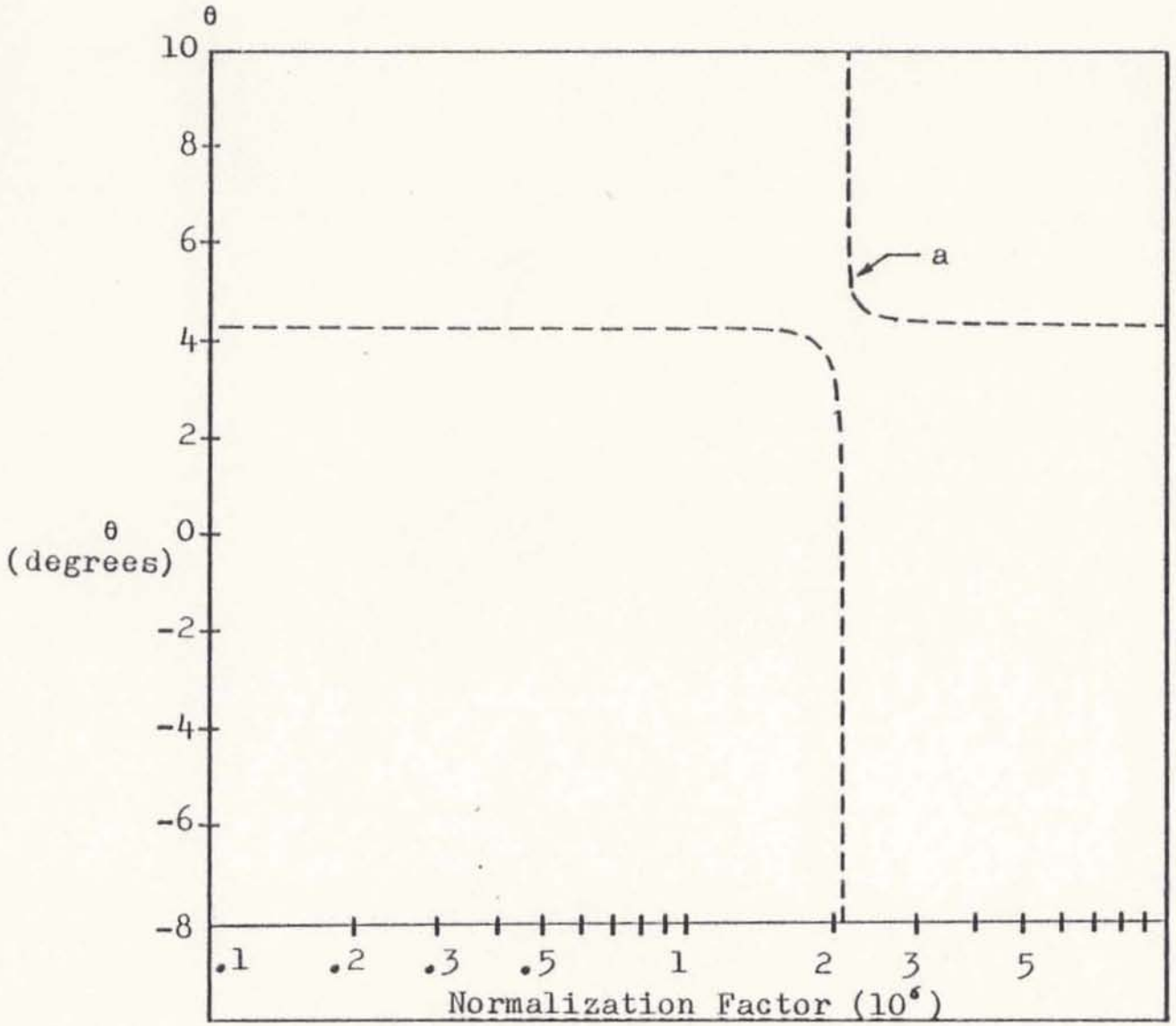


Figure 11. Normalization of $\theta = 7.5°$

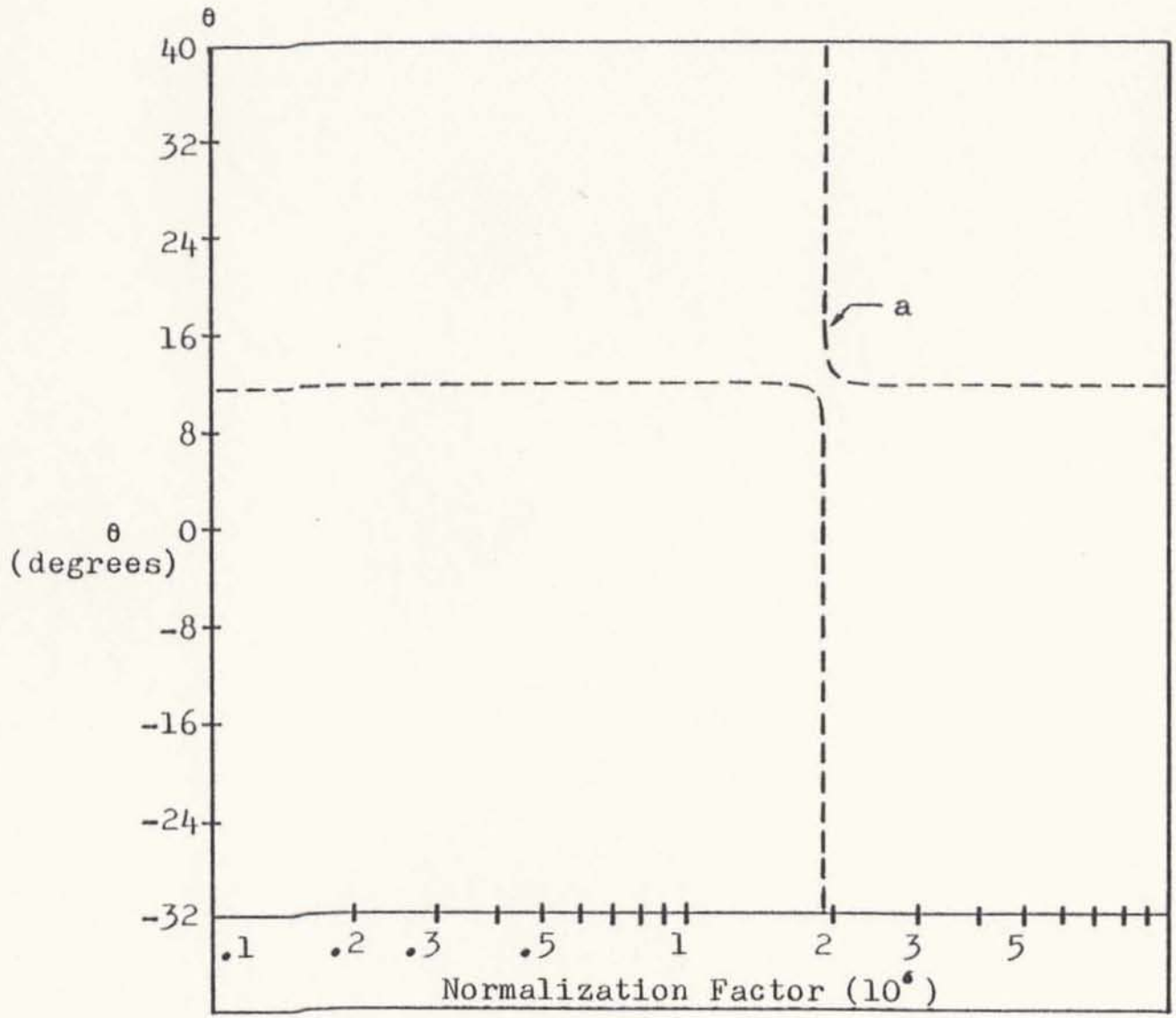


Figure 12. Normalization of $\theta = 22.5°$

CASE #	ORIGINAL BIAS			LEAST SQUARE RESULTS					
	α	β	θ	α	|α| (%) ERROR	β	|β| (%) ERROR	θ	|θ| (%) ERROR
1	0	0	0	0	0	0	0	0	0
2	0	0	2.5	0	0	0	0	1.448	-42.1
3	0	0	5.0	0	0	0	0	3.873	-22.5
4	0	0	7.5	0	0	0	0	6.366	-15.1
5	0	0	10.0	0	0	0	0	8.603	-14.0
6	0	0	22.5	0	0	0	0	16.400	-27.1
7	5.2	2.6	0	5.001	-3.8	2.317	-10.9	0.155	-
8	2.6	5.2	0	2.952	13.5	4.034	-22.4	0.142	-
9	2.6	-5.2	0	2.952	13.5	-4.034	-22.4	0.142	-
10	-5.2	2.6	0	-5.001	-3.8	2.315	-10.9	0.155	-
11	5.2	2.6	2.5	5.634	8.3	1.822	-29.9	1.617	-35.3
12	5.2	2.6	7.5	7.090	36.3	1.509	-42.0	6.660	-11.2
13	2.6	5.2	7.5	4.166	60.2	4.134	-20.3	6.883	-8.2
14	5.2	2.6	22.5	7.125	37.0	0.462	-82.2	16.284	-27.6

Table 2. Least square results for noncloud maps with center element normalization

rotation cases of table 1; however, these results were based on a normalization factor of the center element of the correlation array. In table 2, the average relative rotation error (cases 2 through 6) decreased to 24% from the 40% in table 1. The changes in the translation errors were mixed, but not as significant as the changes in the rotation errors. Since the rotation results seem to contribute the most to the map registration error, a center element normalization appears to be an improvement.

The next investigation regarded the introduction of cloud noise onto the map. A cloud of 100 x 100 pixels was placed on the map centered at the northwest corner of the island as shown in figure 8. At this point, the cloud's affect will be felt during the convolution of both land and sea pixels and since it is at a corner, it will affect the correlation in the northwest differently than in the southeast. Table 3 gives the least square solutions for cloud noise with a normalization factor of the value of the center element of the correlation array.

The first case to be examined is case 15 where no bias was applied to the weather map. In this case, a bias does result due to the cloud noise because the correlation array is no longer symmetrical with respect to the u- and v-axes. Since this lack of symmetry causes an error in

CASE #	ORIGINAL BIAS			LEAST SQUARE RESULTS					
	α	β	θ	α	$\lvert\alpha\rvert$ (%) ERROR	β	$\lvert\beta\rvert$ (%) ERROR	θ	$\lvert\theta\rvert$ (%) ERROR
15	0	0	0	-0.241	-	1.224	-	0.729	-
16	0	0	7.5	0.522	-	0.172	-	6.506	-13.3
17	5.2	2.6	0	4.809	-7.5	3.648	40.3	1.080	-
18	5.2	2.6	7.5	7.503	44.3	1.748	-32.8	7.772	-9.7
19	2.6	5.2	7.5	4.626	77.9	4.385	-15.7	6.832	-8.9
20	5.2	2.6	22.5	7.125	37.0	0.462	-82.2	16.284	-27.6

Table 3. Least square results with cloud noise

the least square result, it can be expected that a map with cloud noise could never be registered exactly.

A comparison of cases 16,18,19 and 20, with cases 4,12,13 and 14 of table 2, shows that the cloud had little affect on the rotational result. However, when the rotational bias approached zero, as seen in case 15 and 17, a rotation error of as much as one degree was introduced by the cloud (the respective cases, 1 and 7, of table 2 produced no more than 0.155° rotation error). It is interesting to note that, in case 20, the cloud had no affect on the translation or rotation error when compared to case 14. This result is due to the fact that, the large rotation placed the cloud entirely over the water area of the data base causing little change in the correlation array.

VII. ITERATIVE REGISTRATION METHOD

From the previous section, it is apparent that a single, one-pass registration using least squares may not be sufficient for proper image registration. With the decreasing cost of random access memory and processing hardware, the iterative approach becomes more economical. In this method, the procedure outlined in this paper would be performed by loading the weather map in one page of memory and performing the correlation with a data base in another page of memory. The shifting operation necessary to obtain the correlation array may be incorporated in the address of the data base pixel being used for comparison. Once the sampled correlation array has been constructed and a least square solution for the translation and rotation has been determined, the map would then be reloaded with an opposite shift and rotation bias applied to it as it is placed in memory from a mass storage device such as a disc drive and another least square registration would be applied to this corrected map.

For example, if a weather map that has rotation only, such as case 6 ($\alpha=\beta=0$, $\theta=22.5°$), is applied to the least square registration method, then the rotation will

have been determined to be (from table 2) 16.4° . The weather map is then reloaded into core with a -16.4° rotational bias added to it changing the actual rotational bias on the weather map to

$$22.5 - 16.4 = 6.1 \text{ degrees.}$$

This new bias is close to case 3 ($\alpha=\beta=0$, $\theta=5°$) which produces a least square estimate (from table 2) of 3.873°. If the weather map is corrected by this new estimate, then the new actual rotational bias would be

$$6.1 - 3.873 = 2.227 \text{ degrees,}$$

which is close to case 2 ($\alpha=\beta=0$, $\theta=2.5°$). Another repetition of this process yields an estimate of 1.448° and if the map is corrected by this amount, then the actual rotational bias would be

$$2.227 - 1.448 = 0.779 \text{ degrees.}$$

Now after three passes, the rotational error with respect to the original bias is

$$\frac{0.779}{22.5} \times 100 = 3.5\%$$

which is considerably less than the error of case 6 in table 2 of 27.1%.

The translational results become more accurate as rotational bias is removed from the weather map. This is seen by examination of cases 7,11,12, and 14 of table 2.

In these cases, the magnitude of the error in α dropped to 3.8% from 37% and the magnitude of the error in β dropped to 10.9% from 82.2% as the rotation decreased from 22.5° to zero.

The size of memory needed to perform this type of registration would depend on the search window size and the size of the data base. The data base area would have to be large enough to accomplish the shift of the weather map when performing a convolution. Two bits per pixel would be required to represent land, sea and cloud pixel values.

VIII. CONCLUSION AND RECOMMENDATIONS

The correlation of irregularly shaped map components produces a system of equations that do not yield an exact solution for the image registration parameters of translation and rotation. The least squares method provides a method for obtaining an approximate solution to these equations. The results of this paper bear this fact out by showing that errors result when a set of equations, derived from rectangular components, is fitted to a conical shape using least squares.

The least square method itself requires very little computer time (the FORTRAN program in appendix 3 required only ten seconds of run time) although much time and/or hardware is needed to generate the correlation arrays used by the least square method. Since this approximation can be improved by using an iterative approach and the sampled correlation time can be improved with an array processor, these two ideas can be combined to speed up the registration process.

There is a need for additional investigation into the problem of cloud noise. Cloud noise caused an increased amount of error as compared to the results of

cases without clouds. It was shown that the presence of a cloud made it impossible to achieve an exact solution because the cloud introduced a zero bias error when no translation or rotation had been applied. Future studies into methods of reducing the effects of cloud noise are needed before the least squares registration method could be applied to applications such as weather analysis.

Also, the problem of normalization could use futher analysis. It was shown that a normalization factor of the value of the correlation array's center element yielded an apparent closer solution. However, there was still a considerable amount of error in those least square solutions. Extremely high or low normalization factors tended to produce a solution of approximately half the desired value which suggests that these factors could be used and their results simply doubled. If this holds true when translation biases are incorporated with rotational biases and in the presence of clouds, then some optimum normalization factor of a high or low extreme may yield a more exact picture registration.

APPENDIX 1. SOLUTION OF REGISTRATION PARAMETERS

The coefficients of equations (9) and (10) of the main text can be written as the following set of equations:

$$t_1 = \cos^2\theta/a^2 + \sin^2\theta/b^2 \tag{20}$$

$$t_2 = \sin^2\theta/a^2 + \cos^2\theta/b^2 \tag{21}$$

$$t_3 = 2\sin\theta\cos\theta[1/a^2 - 1/b^2] \tag{22}$$

$$t_4 = -2[\beta \sin\theta/b^2 + \alpha \cos\theta/a^2] \tag{23}$$

$$t_5 = 2[\alpha \sin\theta/a^2 - \beta \cos\theta/b^2] \tag{24}$$

$$t_6 = \alpha^2/a^2 + \beta^2/b^2 \quad . \tag{25}$$

The coefficients t_1, t_2 and t_3 can be combined,

$$t_3/(t_2 - t_1) = \tan 2\theta \quad , \tag{26}$$

to yield a value for the rotation,

$$\theta = \tfrac{1}{2}\tan^{-1}[t_3/t_2 - t_1)] \quad . \tag{27}$$

For convenience, intermediate values of F and G are defined as

$$F = (t_3/\sin 2\theta) + (t_1 + t_2) = 2/b^2 \tag{28}$$

$$G = (t_3/\sin 2\theta) - (t_1 + t_2) = -2/a^2 \tag{29}$$

which leads to the solution of the elliptical values,

$$a^2 = -2/G \tag{30}$$

$$b^2 = 2/F \quad . \tag{31}$$

These values, along with the coefficients t_4 and t_5, yield the translational components,

$$\beta = -(t_4 \sin\theta + t_5 \cos\theta)/F \tag{32}$$

$$\alpha = (t_4 + \beta F \sin\theta)/G \cos\theta \quad . \tag{33}$$

The results of equations (30,31,32,33) can be checked against t_6 with equation (25). When one subtracts from the coefficient t_6 as follows:

$$t_6 - \frac{(33)^2}{(30)} + \frac{(32)^2}{(31)} = D \quad , \tag{34}$$

the result should ideally equal to zero. In this paper this difference was, in orders of magnitude, 10^{-4} or less. Table 4 shows some representative values for D arrived at in this paper.

α	β	θ	CLOUD	D (10^{-4})
0	0	0	NO	-2.00
0	0	7.5	NO	-1.77
0	0	22.5	NO	-0.23
5.2	2.6	0	NO	-3.81
5.2	2.6	22.5	NO	-0.26
0	0	0	YES	-1.88
0	0	7.5	YES	-0.18
5.2	2.6	22.5	YES	-2.64

Table 4. Check values against coefficient t_6

APPENDIX 2. FORTRAN CORRELATION ARRAY GENERATOR

```
C
C
C     GENERATES ARRAY FROM A HYPOTHETICAL MAP MODEL THAT HAS BEEN
C     SHIFTED AND ROTATED
C
C
C
      DIMENSION XD(4),YD(4),XW(4),YW(4),XC(4),YC(4),IMTCH(5,5)
      WRITE(6,100)
C
  100 FORMAT(1H1,14X,15H THESIS: CORDAN//9X,32H DATA BASE MAP: CENTER AT
     1 ORIGIN/24X,16H 210 PIXELS WIDE/24X,16H 900 PIXELS HIGH/24X,20H SU
     2RROUNDED BY WATER/)
C
  102 FORMAT(1H0,21X,39H CLOUD: 100 X 100 PIXELS, CENTERED AT (,F8.3,3H
     1, ,F8.3,30H) ...BEFORE SHIFT AND ROTATION//)
C
      XCC=-105
      YCC=450
      READ(5,110)ICNT
  110 FORMAT(3X,I2)
      DO 599 MCNT=1,ICNT
C
C     GET SHIFT, ROTATION, CLOUD PARAMETER
C
      READ(5,111)ALP,BET,ROT,ICLD
  111 FORMAT(3X,3(2X,F8.3),6X,I1)
      CLD=FLOAT(ICLD)
      WRITE(6,101)ALP,BET,ROT
  101 FORMAT(1H0,9X,24HWEATHER MAP: CENTER AT (,F7.3,3H , ,F7.3,3H)    /
     121X,11H ROTATION= ,F8.3,10H DEGREES  )
      IF(CLD.EQ.0.)GO TO 10
      WRITE(6,102)XCC,YCC
      GO TO 11
   10 WRITE(6,103)
  103 FORMAT(1H0,21X,10H NO CLOUDS//)
   11 PI=3.141592654
      RAD=ROT*PI/180
      RSIN=SIN(RAD)
      RCOS=COS(RAD)
C
C     DEFINE DATA BASE MAP
C
      XD(1)=105
      XD(2)=-105
      XD(3)=-105
      XD(4)=105
      YD(1)=450
```

```
      YD(2)=450
      YD(3)=-450
      YD(4)=-450
      XDMX=105
      YDMX=450
      WRITE(6,199)
  199 FORMAT(1H1,2X)
      DO 41 J=1,5
      JJJ=6-J
      YSH=FLOAT((3-J)*20)+BET
C
C     DEFINE ISLAND (Y-COORDINATES)
C
      DO 12 K=1,4
   12 YW(K)=XD(K)*RSIN+YD(K)*RCOS+YSH
C
C     DEFINE CLOUD ( Y-COORDINATES )
C
      YC(1)=(XCC+50)*RSIN+(YCC+50)*RCOS+YSH
      YC(2)=(XCC-50)*RSIN+(YCC+50)*RCOS+YSH
      YC(3)=(XCC-50)*RSIN+(YCC-50)*RCOS+YSH
      YC(4)=(XCC+50)*RSIN+(YCC-50)*RCOS+YSH
      YMX=ABS(YW(1))
      DO 16 K=1,4
      AYK=ABS(YW(K))
      IF(YMX-AYK)13,14,14
   13 YMX=AYK
   14 IF(CLD.EQ.0.)GO TO 16
      AYK=ABS(YC(K))
      IF(YMX-AYK)15,16,16
   15 YMX=AYK
   16 CONTINUE
      IF(YMX-YDMX)17,18,18
   17 YMX=YDMX
   18 JYMAX=INT(YMX+.5)
      DO 40 I=1,5
      XSH=FLOAT((I-3)*20)+ALP
      DO 22 K=1,4
C
C     DEFINE ISLAND (X-COORDINATES)
C
   22 XW(K)=XD(K)*RCOS-YD(K)*RSIN+XSH
C
C     DEFINE CLOUD (X-COORDINATES)
C
      XC(1)=(XCC+50)*RCOS-(YCC+50)*RSIN+XSH
      XC(2)=(XCC-50)*RCOS-(YCC+50)*RSIN+XSH
      XC(3)=(XCC-50)*RCOS-(YCC-50)*RSIN+XSH
```

```
      XC(4)=(XCC+50)*RCOS-(YCC-50)*RSIN+XSH
      XMX=ABS(XW(1))
      DO 26 K=1,4
      AXK=ABS(XW(K))
      IF(XMX-AXK)23,24,24
   23 XMX=AXK
   24 IF(CLD.EQ.0.)GO TO 26
      AXK=ABS(XC(K))
      IF(XMX-AXK)25,26,26
   25 XMX=AXK
   26 CONTINUE
      IF(XMX-XDMX)27,28,28
   27 XMX=XDMX
   28 IXMAX=INT(XMX+.5)
      IMTCH(I,JJJ)=2250000-(IXMAX*JYMAX)*4
      WRITE(6,104)XSH,YSH,IXMAX,JYMAX
  104 FORMAT(1H0,8X,9H  SHIFT=(,F8.2,2H ,,F8.2,2H )/1X,17H SCANNING FIEL
     1D=(,I4,3H X ,I4,3H)*4/)
      IF(CLD.EQ.0.)GO TO 30
C
C     DEFINE CLOUD AND ISLAND BOUNDARIES
C
      WRITE(6,105)
  105 FORMAT(5X,13H CLOUD LINES:/)
      CALL PTLINE(XC(4),YC(4),XC(1),YC(1),SCA,YICA,1)
      CALL PTLINE(XC(1),YC(1),XC(2),YC(2),SCB,YICB,1)
      CALL PTLINE(XC(2),YC(2),XC(3),YC(3),SCC,YICC,1)
      CALL PTLINE(XC(3),YC(3),XC(4),YC(4),SCD,YICD,1)
   30 WRITE(6,106)
  106 FORMAT(5X,13H W.MAP LINES:/)
      CALL PTLINE(XW(4),YW(4),XW(1),YW(1),SWA,YIWA,1)
      CALL PTLINE(XW(1),YW(1),XW(2),YW(2),SWB,YIWB,1)
      CALL PTLINE(XW(2),YW(2),XW(3),YW(3),SWC,YIWC,1)
      CALL PTLINE(XW(3),YW(3),XW(4),YW(4),SWD,YIWD,1)
C
C     SCAN SEARCH WINDOW
C
      DO 40 IP=1,IXMAX
      X=FLOAT(IP)-.5
      DO 40 JP=1,JYMAX
      Y=FLOAT(JP)-.5
      IF(X-105)31,31,33
   31 IF(Y-450)32,32,33
   32 N1=1
      N2=1
      N3=1
      N4=1
      GO TO 34
```

```
 33 N1=-1
    N2=-1
    N3=-1
    N4=-1
 34 IF(CLD.EQ.0.)GO TO 35
    CALL CLOUD(SCA,SCB,YICA,YICB,YICC,YICD,X,Y,ROT,N1)
    X=-X
    CALL CLOUD(SCA,SCB,YICA,YICB,YICC,YICD,X,Y,ROT,N2)
    Y=-Y
    CALL CLOUD(SCA,SCB,YICA,YICB,YICC,YICD,X,Y,ROT,N3)
    X=-X
    CALL CLOUD(SCA,SCB,YICA,YICB,YICC,YICD,X,Y,ROT,N4)
    Y=-Y
    IF(N1.EQ.0)GO TO 36
 35 CALL WLAND(SWA,SWB,YIWA,YIWB,YIWC,YIWD,X,Y,ROT,N1)
 36 X=-X
    IF(N2.EQ.0)GO TO 37
    CALL WLAND(SWA,SWB,YIWA,YIWB,YIWC,YIWD,X,Y,ROT,N2)
 37 Y=-Y
    IF(N3.EQ.0)GO TO 38
    CALL WLAND(SWA,SWB,YIWA,YIWB,YIWC,YIWD,X,Y,ROT,N3)
 38 X=-X
    IF(N4.EQ.0)GO TO 39
    CALL WLAND(SWA,SWB,YIWA,YIWB,YIWC,YIWD,X,Y,ROT,N4)
 39 Y=-Y
    IMTCH(I,JJJ)=IMTCH(I,JJJ)+N1+N2+N3+N4
 40 CONTINUE
    WRITE(6,199)
 41 CONTINUE
    WRITE(6,101)ALP,BET,ROT
    IF(CLD.EQ.0.)GO TO 510
    WRITE(6,102)XCC,YCC
    GO TO 511
510 WRITE(6,103)
511 WRITE(6,107)
107 FORMAT(1H0,23H H(I,J); SAMPLE MATRIX //46X,10H H(40,40) /)
    DO 50 J=1,5
    JY=6-J
    WRITE(6,108)(IMTCH(IX,JY),IX=1,5)
108 FORMAT(7X,5(I8,2X))
 50 CONTINUE
    WRITE(6,109)
109 FORMAT(6X,11H H(-40,-40))
    WRITE(6,199)
599 CONTINUE
    END
```

```
      SUBROUTINE PTLINE(X1,Y1,X2,Y2,SLOPE,YINCP,MF)
C
C          DETERMINES THE SLOPE AND Y-INTCPT OF A LINE FORMED
C        BY TWO POINTS (X,Y). WILL PRINT RESULT FOR MF NOT.EQ.0
C
      B=X2-X1
      IF(B)11,10,11
   10 SLOPE=0
      YINCP=X1
      IF(MF)30,99,30
   11 A=Y1-Y2
      SLOPE=-A/B
      YINCP=Y1-X1*SLOPE
      IF(MF.EQ.0)GO TO 99
      WRITE(6,100)X1,Y1,X2,Y2,SLOPE,YINCP
  100 FORMAT(5X,7H PT.1=(,F8.2,2H ,,F8.2,9H ) PT.2=(,F8.2,2H ,,F8.2,2H )
     1,5X,10H LINE: Y=(,E12.4,4H)X+(,F8.2,3H ) )
      GO TO 99
   30 WRITE(6,101)X1,Y1,X2,Y2,YINCP
  101 FORMAT(5X,7H PT.1=(,F8.2,2H ,,F8.2,9H ) PT.2=(,F8.2,2H ,,F8.2,2H )
     1,5X,9H LINE: X=,F8.2)
   99 CONTINUE
      RETURN
      END


      SUBROUTINE WLAND(SA,SB,YIA,YIB,YIC,YID,X,Y,R,NX)
C
C          DETERMINES IF PIXEL IS WITHIN LAND BOUNDARYS
C
      IF(R)20,30,10
   10 BA=Y-SA*X
      BB=Y-SB*X
      IF(BA.GT.YIA)GO TO 99
      IF(BA.LT.YIC)GO TO 99
      IF(BB.GT.YIB)GO TO 99
      IF(BB.LT.YID)GO TO 99
      GO TO 98
   20 BA=Y-SA*X
      BB=Y-SB*X
      IF(BA.LT.YIA)GO TO 99
      IF(BA.GT.YIC)GO TO 99
      IF(BB.GT.YIB)GO TO 99
      IF(BB.LT.YID)GO TO 99
      GO TO 98
   30 IF(X.GT.YIA)GO TO 99
      IF(X.LT.YIC)GO TO 99
      IF(Y.GT.YIB)GO TO 99
      IF(Y.LT.YID)GO TO 99
   98 CONTINUE
      RETURN
   99 NX=NX*(-1)
      RETURN
      END
```

```
      SUBROUTINE CLOUD(SCA,SCB,YICA,YICB,YICC,YICD,X,Y,R,NX)
C
C          DETERMINES IF PIXEL IS WITHIN A CLOUD.
C
      IF(R)20,30,10
   10 BA=Y-SCA*X
      BB=Y-SCB*X
      IF(BA.GT.YICA)GO TO 99
      IF(BB.LT.YICD)GO TO 99
      IF(BA.LT.YICC)GO TO 99
      IF(BB.GT.YICB)GO TO 99
      GO TO 98
   20 BA=Y-SCA*X
      BB=Y-SCB*X
      IF(BA.LT.YICA)GO TO 99
      IF(BB.LT.YICD)GO TO 99
      IF(BA.GT.YICC)GO TO 99
      IF(BB.GT.YICB)GO TO 99
      GO TO 98
   30 IF(X.GT.YICA)GO TO 99
      IF(Y.LT.YICD)GO TO 99
      IF(X.LT.YICC)GO TO 99
      IF(Y.GT.YICB)GO TO 99
   98 NX=0
      RETURN
   99 CONTINUE
      RETURN
      END
```

APPENDIX 3. FORTRAN LEAST SQUARES PROGRAM

```
C           THESIS:LEAST SQ.APPROX FROM ELLIPTICAL CONE DATA...CORDAN
C
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)
      DIMENSION H(5,5),SMH(25),AM(25,6),A(6,6),AA(6,6),T(6),B(5),AAA(6,6
     1),S(6)
      PI=3.141592654
      READ(5,110)ICNT
  110 FORMAT(8X,I2)
      DO 999 MCNT=1,ICNT
      WRITE(6,100)
  100 FORMAT(1H1,15X,54HTHESIS: LEAST SQ. FIT OF ELLIPTICAL CONE DATA...
     1CORDAN//)
C
C      GET SAMPLED CONE DATA H(I,J)
C
      DO 5 J=1,5
      READ(5,101)(H(I,J),I=1,5)
  101 FORMAT(1X,5F10.0)
    5 CONTINUE
      WRITE(6,149)
  149 FORMAT(5X,7H H(I,J)//)
      WRITE(6,150)((H(I,J),I=1,5),J=1,5)
  150 FORMAT((5(E15.7,2X)))
      HMAX=H(3,3)
      DO 10 J=1,5
      BJ=FLOAT(J)
      YB=-40+(BJ-1)*20
      DO 10 I=1,5
      AI=FLOAT(I)
      XA=-40+(AI-1)*20
C
C      GENERATE 25 EQUATIONS FROM CONE DATA
C
      K=5*(J-1)+I
      SMH(K)=((HMAX-H(I,J))/HMAX)**2.
      AM(K,1)=XA*XA
      AM(K,2)=YB*YB
      AM(K,3)=XA*YB
      AM(K,4)=XA
      AM(K,5)=YB
      AM(K,6)=1.
   10 CONTINUE
C
C      A=A TRANSPOSE*A MATRIX
```

```
      DO 11 J=1,6
      DO 11 I=1,6
      A(I,J)=0.
      DO 11 K=1,25
      A(I,J)=A(I,J)+AM(K,I)*AM(K,J)
   11 CONTINUE
      CALL DMINVR(A,DETM,MS,AA)
      DO 21 I=1,6
      DO 21 J=1,6
      IF(J-I)22,21,21
   22 AA(J,I)=AA(I,J)
   21 CONTINUE
C
C      S(I)=A TRANSPOSE*(N-H(I,J))**2
C
      DO 14 I=1,6
      S(I)=0.
      DO 13 J=1,25
   13 S(I)=S(I)+AM(J,I)*SMH(J)
   14 CONTINUE
C
C      THE COEFFICIENTS T(I)= A INVERSE*S(I)
C
      DO 16 I=1,6
      T(I)=0.
      DO 16 J=1,6
   16 T(I)=T(I)+AA(I,J)*S(J)
      WRITE(6,116)HMAX
  116 FORMAT(1H0,2X,19H T(1),T(2),...,T(6),10H     HMAX=,F9.0//)
      WRITE(6,117)(T(I),I=1,6)
  117 FORMAT(2X,6(E14.6,2X)//)
C
C      CALC. SHIFT AND ROTATION FROM T(I)'S
C
      IF(T(2)-T(1)) 52,53,52
   53 WRITE(6,156)
  156 FORMAT(1H0,5X,41H T(1)=T(2)...ASSUME ROT=45,SIGN=SIGN T(3)//)
      IF(T(3))54,55,55
   54 THETA=ATAN(-1.)
      GO TO 60
   55 THETA=ATAN(1.)
      GO TO 60
   52 IF(T(3)) 57,56,57
   56 WRITE(6,157)
  157 FORMAT(1H0,5X,22H T(3)=0...ASSUME ROT=0//)
   58 THETA=0.
      TAU=T(2)-T(1)
      GO TO 62
   57 THETA=ATAN(T(3)/(T(2)-T(1)))/2.
   60 TAU=T(3)/SIN(2.*THETA)
```

```
 62 THDEG=THETA*180./PI
    F=TAU+T(1)+T(2)
    G=TAU-T(1)-T(2)
    BETA=(-1.)*(T(4)*SIN(THETA)+T(5)*COS(THETA))/F
    ALPHA=(T(4)+BETA*F*SIN(THETA))/(G*COS(THETA))
    CKDIF=T(6)-(F*BETA**2.-G*ALPHA**2.)/2.
    WRITE(6,118)ALPHA,BETA,THDEG,CKDIF
118 FORMAT(1H0,2X,6HALPHA=,F7.3,7H  BETA=,F7.3,8H  THETA=,F8.3,13H  CH
   1ECK DIFF=,E14.6//)
999 CONTINUE
    END
```

```
      SUBROUTINE DMINVR(A,DET,MSF,TF)
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)
      DOUBLE PRECISION A,DET,TF
      DIMENSION A(6,6),TF(6,6),P(6,6)
C      A=INVERSE A (GAUSE JORDAN REDUCTION)
      DO 20 I=1,6
      DO 20 J=1,6
   20 P(I,J)=A(I,J)
      DET =1.
      DO 23 I=1,6
      DO 23 J=1,6
      IF(I-J)22,21,22
   21 TF(I,J)=1.
      GO TO 23
   22 TF(I,J)=0.
   23 CONTINUE
      DO 550 I=1,6
C      SEARCH PIVOT ELEM.
      AMAX=0.
   24 DO 27 J=1,6
      AB=P(J,I)
   25 IF(AMAX-DABS(AB))26,27,27
   26 IROW=J
      ICOL=I
      AMAX=DABS(AB)
   27 CONTINUE
      IF(AMAX)28,88,28
C      INTERCHANGE ROWS TO PIVOT ELEM.ON DIAGONAL
   28 IF(IROW-ICOL)29,260,29
   29 CONTINUE
      DO 200 L=1,6
      SWAP=P(IROW,L)
      P(IROW,L)=P(I,L)
  200 P(I,L)=SWAP
      DO 215 L=1,6
      SWAP=TF(IROW,L)
      TF(IROW,L)=TF(I,L)
  215 TF(I,L)=SWAP
  260 CONTINUE
      PIVOT=P(I,I)
      DET=DET*PIVOT
C      DIVIDE PIVOT ROW BY PIVOT ELEM.
      DO 350 L=1,6
  350 P(I,L)=P(I,L)/PIVOT
      DO 355 L=1,6
  355 TF(I,L)=TF(I,L)/PIVOT
C      REDUCE NON-PIVOT ROWS
      DO 550 K=1,6
```

```
      IF(K-ICOL)400,550,400
  400 TP=P(K,I)
      P(K,I)=0.
      DO 450 L=1,6
  450 P(K,L)=P(K,L)-P(I,L)*TP
      DO 455 L=1,6
  455 TF(K,L)=TF(K,L)-TF(I,L)*TP
      MSF=0
      GO TO 550
   88 MSF=1
  550 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE DMPAB(A,B,C,IWRT)
      DOUBLE PRECISION A(6,6),B(6,6),C(6,6)
C
C       A*B=C ,,, MATRIX MULTIPLICATION
C       AAA(I,J)=A INVERSE*A, THIS IS A CHECK FOR ERROR IN THE INVERSE
      DO 6 I=1,6
      DO 6 J=1,6
      C(I,J)=0.0
      DO 6 K=1,6
    6 C(I,J)=C(I,J)+A(I,K)*B(K,J)
      RETURN
      END
```

APPENDIX 4. DATA

```
MAP MODEL CENTER: (  0.0 ,  0.0 )
       ROTATION:   0.0 DEGREES

NO CLOUD


R(U,V):
                                                          R(40,40)

 2078600.   2147600.   2216400.   2147600.   2078600.
 2092400.   2162600.   2233200.   2162600.   2092400.
 2106000.   2178000.   2250000.   2178000.   2106000.
 2092400.   2162600.   2233200.   2162600.   2092400.
 2078600.   2147600.   2216400.   2147600.   2078600.

R(-40,-40)

          LEAST SQUARE SOLUTION

NORMALIZATION CONSTANT = 2250000

     ALPHA=  0.000     BETA=  0.000     THETA=  0.000


NORMALIZATION CONSTANT =  2250000.

     ALPHA=  0.000     BETA=  0.000     THETA=  0.000
```

```
MAP MODEL CENTER: (  0.0 ,  0.0 )
       ROTATION:   2.5 DEGREES

NO CLOUD


R(U,V):
                                                          R(40,40)

 2076680.   2143248.   2184392.   2150360.   2061480.
 2091736.   2161924.   2199736.   2164324.   2093624.
 2104956.   2176724.   2213436.   2176724.   2104956.
 2093624.   2164324.   2199736.   2161924.   2091736.
 2061480.   2150360.   2184392.   2143248.   2076680.

R(-40,-40)


          LEAST SQUARE SOLUTION

NORMALIZATION CONSTANT = 2250000

      ALPHA=  0.000     BETA=  0.000     THETA=  1.330


NORMALIZATION CONSTANT =  2213436.

      ALPHA=  0.000     BETA=  0.000     THETA=  1.448
```

```
MAP MODEL CENTER: (  0.0 ,  0.0 )
       ROTATION:   5.0 DEGREES

NO CLOUD


R(U,V):

                                                          R(40,40)

 2070072.   2131968.   2162928.   2137172.   2084736.
 2091644.   2147436.   2166816.   2149460.   2095424.
 2104364.   2159592.   2178024.   2159592.   2104364.
 2095424.   2149460.   2166816.   2147436.   2091644.
 2084736.   2137172.   2162928.   2131968.   2070072.

R(-40,-40)


          LEAST SQUARE SOLUTION

NORMALIZATION CONSTANT = 2250000

      ALPHA=  0.000     BETA=  0.000     THETA=  3.174


NORMALIZATION CONSTANT =  2178024.

      ALPHA=  0.000     BETA=  0.000     THETA=  3.873
```

```
MAP MODEL CENTER: (  0.0 ,  0.0 )
       ROTATION:   7.5 DEGREES

NO CLOUD


R(U,V):

                                                          R(40,40)

 2068072.   2107256.   2121912.   2112064.   2077688.
 2083720.   2121304.   2134360.   2122912.   2086936.
 2094140.   2131256.   2143592.   2131256.   2094140.
 2086936.   2122912.   2134360.   2121304.   2083720.
 2077688.   2112064.   2121912.   2107256.   2068072.

R(-40,-40)

          LEAST SQUARE SOLUTION

NORMALIZATION CONSTANT = 2250000

      ALPHA=  0.000     BETA=  0.000     THETA=  4.876


NORMALIZATION CONSTANT =  2143592.

      ALPHA=  0.000     BETA=  0.000     THETA=  6.633
```

```
MAP MODEL CENTER: (  0.0 ,  0.0 )
       ROTATION:  10.0 DEGREES

NO CLOUD


R(U,V):

                                                          R(40,40)

 2049832.   2079740.   2091228.   2084292.   2058932.
 2064036.   2092344.   2102232.   2093696.   2066736.
 2072512.   2100568.   2107924.   2100568.   2072512.
 2066736.   2093696.   2102232.   2092344.   2064636.
 2058932.   2084292.   2091228.   2079740.   2049832.

R(-40,-40)

          LEAST SQUARE SOLUTION

NORMALIZATION CONSTANT = 2250000

     ALPHA=  0.000    BETA=  0.000    THETA=  6.356


NORMALIZATION CONSTANT =  2109924.

     ALPHA=  0.000    BETA=  0.000    THETA=  9.603
```

```
MAP MODEL CENTER: (  0.0 ,  0.0 )
       ROTATION:  22.5 DEGREES

NO CLOUD


R(U,V):
                                                          R(40,40)

  1916662.   1931540.   1938429.   1936716.   1927022.
  1923594.   1936716.   1942004.   1938692.   1927396.
  1927203.   1940008.   1944524.   1940008.   1927203.
  1927396.   1938692.   1942004.   1936716.   1923594.
  1927022.   1936716.   1938429.   1931540.   1916662.

R(-40,-40)

          LEAST SQUARE SOLUTION

NORMALIZATION CONSTANT = 2250000

      ALPHA=  0.000     BETA=  0.000     THETA= 11.874


NORMALIZATION CONSTANT =  1944524.

      ALPHA=  0.000     BETA=  0.000     THETA= 16.400
```

```
MAP MODEL CENTER: (  5.2 ,  2.6 )
      ROTATION:   0.0 DEGREES

NO CLOUD


R(U,V):
                                                        R(40,40)

 2098100.   2167140.   2201660.   2132620.   2060580.
 2112100.   2182740.   2216060.   2147420.   2076780.
 2121900.   2193660.   2229540.   2157780.   2086020.
 2107300.   2178060.   2213140.   2142980.   2072820.
 2093900.   2162460.   2196740.   2123180.   2050620.

R(-40,-40)

          LEAST SQUARE SOLUTION

NORMALIZATION CONSTANT = 2250000

     ALPHA=  4.780    BETA=  2.300    THETA=  0.122


NORMALIZATION CONSTANT =  2229540.

     ALPHA=  5.001    BETA=  2.317    THETA=  0.155
```

```
MAP MODEL CENTER: (  2.6 ,  5.2 )
       ROTATION:   0.0 DEGREES

NO CLOUD


R(U,V):

                                                          R(40,40)

 2092560.   2161780.   2210220.   2141020.   2071820.
 2106420.   2177320.   2226780.   2155980.   2085180.
 2113340.   2184940.   2233060.   2163460.   2091860.
 2099500.   2169500.   2218500.   2148500.   2078500.
 2085660.   2154060.   2201940.   2133540.   2065140.

R(-40,-40)


          LEAST SQUARE SOLUTION

NORMALIZATION CONSTANT = 3250000

     ALPHA=  2.848    BETA=  4.005    THETA=  0.119


NORMALIZATION CONSTANT =  3235060.

     ALPHA=  2.958    BETA=  4.034    THETA=  0.148
```

```
MAP MODEL CENTER: (  2.6 , -5.2 )
       ROTATION:   0.0 DEGREES

NO CLOUD


R(U,V):
                                                          R(40,40)

 2085660.   2154060.   2201940.   2130540.   2065140.
 2099500.   2163500.   2218500.   2146500.   2078500.
 2113340.   2184940.   2235060.   2162460.   2091860.
 2106420.   2177220.   2226780.   2153980.   2085160.
 2092560.   2161750.   2210220.   2141020.   2071820.

R(-40,-40)

          LEAST SQUARE SOLUTION

NORMALIZATION CONSTANT = 2250000

      ALPHA=  2.348    BETA= -4.005    THETA=  0.119


NORMALIZATION CONSTANT =  2205060.

      ALPHA=  2.352    BETA= -4.034    THETA=  0.142
```

```
MAP MODEL CENTER: ( -5.2 ,  2.6 )
       ROTATION:   0.0 DEGREES

NO CLOUD


R(U,V):
                                                          R(40,40)

 2063360.   2132620.   2201660.   2167140.   2098100.
 2076760.   2147420.   2218060.   2182740.   2112100.
 2086020.   2157760.   2229540.   2193660.   2121900.
 2078320.   2148960.   2219140.   2178060.   2107300.
 2039620.   2128160.   2196740.   2162460.   2093900.

R(-40,-40)


          LEAST SQUARE SOLUTION

NORMALIZATION CONSTANT = 2350000

      ALPHA= -4.780     BETA=  2.300     THETA=  0.122


NORMALIZATION CONSTANT =  2229540.

      ALPHA= -5.001     BETA=  2.317     THETA=  0.155
```

```
MAP MODEL CENTER: (  5.2 ,  2.6 )
      ROTATION:   2.5 DEGREES

NO CLOUD


R(U,V):                                                    R(40,40)

 2076508.   2164328.   2134596.   2134336.   2063056.
 2112012.   2180644.   2199512.   2147652.   2076772.
 2122536.   2132576.   2210324.   2157512.   2085928.
 2110380.   2178740.   2194872.   2141544.   2071360.
 2097620.   2164380.   2179112.   2125300.   2056916.

R(-40,-40)

          LEAST SQUARE SOLUTION

NORMALIZATION CONSTANT = 2350000

     ALPHA=  5.207    BETA=  1.811    THETA=  1.420


NORMALIZATION CONSTANT =  2210324.

     ALPHA=  5.634    BETA=  1.822    THETA=  1.617
```

```
MAP MODEL CENTER: (  5.2 ,  2.6 )
       ROTATION:   7.5 DEGREES

NO CLOUD


R(U,V):

                                                          R(40,40)

 2082508.   2115200.   2123264.   2106644.   2063892.
 2097612.   2128804.   2130288.   2117248.   2074696.
 2103964.   2136824.   2142560.   2123764.   2080244.
 2097412.   2126796.   2131672.   2112020.   2067860.
 2067724.   2115508.   2116764.   2097532.   2051772.

R(-40,-40)

          LEAST SQUARE SOLUTION

NORMALIZATION CONSTANT = 2250000

     ALPHA=  5.858    BETA=  1.611    THETA=  4.978


NORMALIZATION CONSTANT =  2142560.

     ALPHA=  7.090    BETA=  1.509    THETA=  6.660
```

```
MAP MODEL CENTER: (  2.6 ,  5.2 )
       ROTATION:   7.5 DEGREES

NO CLOUD


R(U,V):

                                                          R(40,40)

 2076586.   2114164.   2125220.   2111764.   2073784.
 2056972.   2127948.   2137404.   2122048.   2082766.
 2099036.   2133412.   2142556.   2126376.   2086668.
 2070568.   2122932.   2130780.   2114112.   2072924.
 2061056.   2111620.   2118068.   2099800.   2057012.

R(-40,-40)


          LEAST SQUARE SOLUTION

NORMALIZATION CONSTANT = 2250000

     ALPHA=  3.302    BETA=  3.977    THETA=  4.975


NORMALIZATION CONSTANT =  2142556.

     ALPHA=  4.166    BETA=  4.143    THETA=  6.283
```

```
MAP MODEL CENTER: (  5.2 ,  2.6 )
       ROTATION:  22.5 DEGREES

NO CLOUD


R(U,V):

                                                          R(40,40)

 1922066.   1934778.   1939286.   1935394.   1923106.
 1923282.   1939470.   1942358.   1936666.   1923108.
 1931298.   1941990.   1944134.   1937270.   1922734.
 1931134.   1940234.   1940950.   1933286.   1918598.
 1930298.   1937738.   1936874.   1927610.   1911294.

R(-40,-40)

          LEAST SQUARE SOLUTION

NORMALIZATION CONSTANT = 2250000

      ALPHA=  5.552     BETA=  0.744     THETA= 11.879


NORMALIZATION CONSTANT =  1944134.

      ALPHA=  7.125     BETA=  0.462     THETA= 16.384
```

```
MAP MODEL CENTER: (  0.0 ,  0.0 )
       ROTATION:   0.0 DEGREES

CLOUD: 100 X 100 PIXELS


R(U,V):                                                  R(40,40)

 2073600.   2142000.   2210400.   2145200.   2080000.
 2086800.   2156000.   2225200.   2157600.   2090000.
 2100000.   2170000.   2240000.   2170000.   2100000.
 2086800.   2156000.   2225200.   2156000.   2086800.
 2073600.   2142000.   2210400.   2142000.   2073600.

R(-40,-40)

LEAST SQUARE SOLUTION WITH
NORMALIZATION CONSTANT = 2249000

      ALPHA= -0.234    BETA=  1.191    THETA=  0.708


NORMALIZATION CONSTANT =  2240000.

      ALPHA= -0.241    BETA=  1.224    THETA=  0.729
```

```
MAP MODEL CENTER: (  0.0 ,  0.0 )
       ROTATION:   7.5 DEGREES

CLOUD: 100 X 100 PIXELS


R(U,V):
                                                          R(40,40)

 2063075.   2102259.   2116915.   2106405.   2072031.
 2078723.   2116307.   2129363.   2117021.   2080385.
 2089143.   2126239.   2138099.   2124785.   2086427.
 2081939.   2117915.   2129363.   2114997.   2079641.
 2072691.   2107067.   2116915.   2101517.   2061761.

R(-40,-40)

LEAST SQUARE SOLUTION WITH
NORMALIZATION CONSTANT = 2249000

      ALPHA=  0.427    BETA=  0.120    THETA=  4.989


NORMALIZATION CONSTANT =  2138595.

      ALPHA=  0.522    BETA=  0.172    THETA=  6.506
```

```
MAP MODEL CENTER: (  5.2 ,  2.6 )
       ROTATION:   0.0 DEGREES

CLOUD: 100 X 100 PIXELS


R(U,V):                                                    R(40,40)

 2092710.   2161230.   2196230.   2130670.   2068110.
 2106110.   2175430.   2210430.   2142470.   2074510.
 2115490.   2183370.   2220310.   2150430.   2080350.
 2102090.   2171170.   2205710.   2136630.   2067050.
 2068690.   2136970.   2171110.   2102830.   2034550.

R(-40,-40)

LEAST SQUARE SOLUTION WITH
NORMALIZATION CONSTANT = 2249000

      ALPHA=  4.581    BETA=  3.528    THETA=  0.946


NORMALIZATION CONSTANT =  2220310.

      ALPHA=  4.809    BETA=  3.648    THETA=  1.080
```

```
MAP MODEL CENTER: (  5.2 ,  2.6 )
       ROTATION:   7.5 DEGREES

CLOUD: 100 X 100 PIXELS


R(U,V):

                                                          R(40,40)

  2077605.   2110197.   2118193.   2101177.   2060227.
  2072809.   2123801.   2130197.   2111105.   2067817.
  2100961.   2131621.   2137469.   2117041.   2071985.
  2092409.   2121795.   2126581.   2103393.   2059497.
  2062721.   2110505.   2113701.   2091637.   2045001.

R(-40,-40)


LEAST SQUARE SOLUTION WITH
NORMALIZATION CONSTANT = 2249000

      ALPHA=  6.225    BETA=  1.771    THETA=  5.081


NORMALIZATION CONSTANT =  2137469.

      ALPHA=  7.503    BETA=  1.748    THETA=  6.772
```

```
MAP MODEL CENTER: (  2.6 ,  5.2 )
       ROTATION:   7.5 DEGREES

CLOUD: 100 X 100 PIXELS


R(U,V):

                                                        R(40,40)

 2073093.   2109169.   2120221.   2106055.   2068017.
 2068977.   2122953.   2132405.   2116255.   2075817.
 2094541.   2128417.   2137557.   2120393.   2078427.
 2085373.   2117937.   2125781.   2107745.   2064979.
 2076061.   2106825.   2113069.   2094135.   2050569.

R(-40,-40)


LEAST SQUARE SOLUTION WITH
NORMALIZATION CONSTANT = 2249000

      ALPHA=  3.690     BETA=  4.150     THETA=  4.943


NORMALIZATION CONSTANT =  2137557.

      ALPHA=  4.626     BETA=  4.385     THETA=  6.832
```

```
MAP MODEL CENTER: (  5.2 ,  3.6 )
       ROTATION:  22.5 DEGREES

CLOUD: 100 X 100 PIXELS


R(U,V):

                                                        R(40,40)

  1917066.   1923736.   1934256.   1930334.   1918106.
  1920232.   1934470.   1937308.   1931866.   1918138.
  1926236.   1936990.   1939134.   1932270.   1917734.
  1926134.   1935234.   1935930.   1926236.   1910508.
  1920278.   1932706.   1931674.   1922610.   1906234.

R(-40,-40)


LEAST SQUARE SOLUTION WITH
NORMALIZATION CONSTANT = 2249000

      ALPHA=  5.521     BETA=  0.744     THETA= 11.877


NORMALIZATION CONSTANT =  1939134.

      ALPHA=  7.125     BETA=  0.462     THETA= 16.284
```

APPENDIX 5. NORMALIZATION DATA

For θ equal 2.5 degrees.

Pole location: ~ 2137500

Distance from Pole (E+06)	θ	t_1	t_2	t_3	t_6
-2.0	1.06	-2.57E-02	-6.16E-03	-7.22E-04	4.41E+02
-1.0	1.05	-1.08E-04	-2.60E-05	-2.99E-06	1.00
-0.1	0.77	-3.08E-06	-7.76E-07	-6.21E-08	6.16E-03
-0.05	0.47	-1.48E-06	-3.92E-07	-1.78E-08	2.61E-03
-0.0075	-5.96	-1.64E-07	-7.56E-08	1.87E-08	6.32E-04
POLE	-	-	-	-	-
0.0075	3.80	2.50E-07	2.39E-08	3.01E-08	2.05E-04
0.05	1.61	1.42E-06	3.06E-07	6.27E-08	-4.44E-08
0.1	1.53	1.67E-06	3.66E-07	6.96E-08	-4.67E-04
1.0	1.10	1.27E-05	3.01E-06	3.72E-07	-8.37E-02
2.0	1.09	1.47E-05	3.52E-06	4.27E-07	2.14E-01

t_4 and t_5 are equal to zero.

For θ equal 7.5 degrees.

Pole location: = 2102500

Distance from Pole (E+06)	θ	t_1	t_2	t_3	t_6
-2.0	4.25	-1.23E-02	-4.86E-03	-1.11E-03	4.16E+02
-1.0	4.21	-5.09E-05	-2.01e-05	-4.55E-06	8.95E-01
-0.1	3.45	-1.60E-06	-6.44E-07	-1.15E-07	4.76E-03
-0.05	2.63	-7.88E-07	-3.24E-07	-4.28E-08	1.77E-03
-0.0075	-7.37	-1.25E-07	-6.22E-08	1.66E-08	2.94E-04
POLE	-	-	-	-	-
0.0075	14.62	8.34E-08	2.03E-08	3.53E-08	-3.28E-05
0.05	5.92	6.75E-07	2.54E-07	8.82E-08	-1.34E-04
0.1	5.17	1.22E-06	4.69E-07	1.37E-07	5.52E-04
1.0	4.38	6.36E-06	2.51E-06	5.94E-07	9.56E-02
2.0	4.34	7.29E-06	2.87E-06	6.74E-07	2.28E-01

t_4 and t_5 are equal to zero.

For θ equal 22.5 degrees.

Pole location: ≈ 1932500

Distance from Pole (E+06)	θ	t_1	t_2	t_3	t_6
-1.0	11.60	-2.64E-05	-8.37E-06	-7.74E-06	1.34
-0.1	11.15	-8.43E-07	-2.67E-07	-2.36E-07	6.34E-03
-0.05	10.34	-3.03E-07	-9.58E-08	-7.81E-08	1.12E-03
-0.0075	0.20	-3.65E-08	-1.15E-08	-1.77E-10	7.05E-05
POLE	-	-	-	-	-
0.0075	18.59	4.68E-08	1.48E-08	2.42E-08	-1.94E-05
0.05	12.89	2.82E-07	8.93E-08	9.30E-08	4.12E-04
0.1	12.34	4.96E-07	1.57E-07	1.56E-07	1.79E-03
1.0	11.73	2.39E-06	7.60E-07	7.09E-07	1.08E-01

t_4 and t_5 are equal to zero.

REFERENCES

1. Haralick, R.M. "Automatic Remote Sensor Image Processing." In Digital Picture Analysis, pp. 16-19. Edited by A. Rosenfeld. New York: Springer-Verlag, 1976.

2. Stockham, Thomas G. "High-speed Convolution and Correlation with Applications to Digital Filtering." In Digital Processing of Signals, pp. 203-217. Edited by B. Gold and C.M. Rader. New York: McGraw-Hill, 1969.

3. Anuta, P.F. "Spatial Registration of Multispectral and Multitemporal Digital Imagery Using Fast Fourier Transform Techniques." IEEE Transactions on Geoscience Electronics 8 (October 1970): 353-368.

4. Kunt M. "A Statistical Model for Correlation Functions of Two-Level Digital Facsimilies." Proceedings of the IEEE 63 (February 1975): 327-329.

5. Barnea, Daniel L., and Silverman, Harvey F. "A Class of Algorithms for Fast Digital Image Registration." IEEE Transactions on Computers 21 (February 1972): 179-186.

6. Webber, W.F. "Techniques for Image Registration." Machine Processing of Remotely Sensed Data. Lafayette, Indianna: Purdue University, 1973, p. 1B-1.

7. Pratt, William K. "Correlation Techniques of Image Registration." IEEE Transactions on Aerospace and Electronic Systems 10 (May 1974): 353-358.

8. Gonzalez, Rafael C., and Wintz, Paul. Digital Image Processing. Reading, Massachusetts: Addison-Wesley, 1977.

9. Hunt, B.R. "Digital Image Processing." Proceedings of the IEEE 63 (April 1975): 693-708.

10. Goodman, J.W. Introduction to Fourier Optics. New York: McGraw-Hill, 1968.

11. Rosenfeld, Azriel, and Kak, Avinash C. Digital Picture Processing. New York: Academic Press, 1976, p. 195.

12. Bogdanowicz, J.F. "Preliminary Design of a Partitionable Multi-Microprogrammable Microprocessor System for Image Processing." Lafayette, Indiana: Purdue University, 1977.

13. Rohrbacher, D., and Potter, J.L. "Image Processing with the Staran Parallel Computer." Computer 10 (August 1977): 54-59.

14. Gambino, Lawrence A., and Schrock, Bryce L. "An Experimental Digital Interactive Facility." Computer 10 (August 1977): 22-28.

15. Franklin, Joel N. Matrix Theory. Englewood Cliffs, NJ: Prentice-Hall, 1968.

16. Schwarz, M.R.; Rustishauser, H.; and Stiefel, E. Numerical Analysis of Symmetric Matrices. Englewood Cliffs, NJ: Prentice-Hall, 1973.